

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## FYZIKÁLNÍ SIMULACE V GRAFICKÉ SCÉNĚ

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

BC. MARIÁN JAVORKA

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# FYZIKÁLNÍ SIMULACE V GRAFICKÉ SCÉNĚ

PHYSICAL SIMULATION IN GRAPHICS SCENE

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

BC. MARIÁN JAVORKA

VEDOUcí PRÁCE  
SUPERVISOR

ING. JAN PEČIVA, PH.D.

BRNO 2011

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2010/2011

**Zadání diplomové práce**

Řešitel: **Javorka Marián, Bc.**

Obor: Počítačová grafika a multimédia

Téma: **Fyzikální simulace v grafické scéně**  
**Physical Simulation in Graphics Scene**

Kategorie: Počítačová grafika

Pokyny:

1. Nastudujte si grafickou knihovnu OpenSceneGraph a knihovnu pro fyzikální simulaci Bullet.
2. Na základě předchozího studia navrhnete aplikaci pro fyzikální simulaci aut a grafickou vizualizaci simulace. Fyzikální simulace by měla řešit interakci aut s terénem, mezi sebou a s vybranými objekty scény.
3. Aplikaci implementujte a demonstруйте na jednoduché scéně, ve které bude probíhat fyzikální simulace. Volitelně může mít aplikace formu hry.
4. Vyhodnoťte zkušenosti s aplikací a implementujte vylepšení dle domluvy s vedoucím.
5. Aplikaci prezentujte formou videa či plakátu na UPGM a publikujte na internetu jako open-source.

Literatura:

- Materiály ke knihovnám OpenSceneGraph a Bullet dostupné na internetu.

Při obhajobě semestrální části diplomového projektu je požadováno:

- Jednoduchá funkční simulace.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešení problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Pečiva Jan, Ing., Ph.D.,** UPGM FIT VUT

Datum zadání: 20. září 2010

Datum odevzdání: 25. května 2011

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
602 00 Brno, Božetěchova 2  
L.S.



doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Tento diplomový projekt se zabývá problematikou fyzikální simulace aut. Program je vytvořen v programovacím jazyce C++ za pomoci knihoven OpenSceneGraph a Bullet, které jsou v práci v krátkosti představeny. Celá aplikace má formu jednoduché závodní hry pro jednoho, nebo dva hráče. Pomocí jednoduchého menu si uživatel může zvolit auto a počasí, které výrazně vplývá na jízdné vlastnosti vozidel. Jednou z možností volby počasí je náhodné, kdy dochází k dynamické změně počasí během hry. V aplikaci se zohledňují vzájemné kolize aut spolu s kolizemi s terénem a bariérami na závodním okruhu.

## Abstract

This master's thesis deals with physical simulation of cars. Program is implemented in C++ using OpenSceneGraph and Bullet libraries, which are shortly introduced. The application is implemented as simple racing game for one or two players. Using a simple menu, the user can select a car and the weather, which dramatically affects the handling characteristics of vehicles. Setting weather to random is one of the options leading to a dynamic weather changes during the game. The application takes into account collisions among cars and collisions with the terrain and bars on a racing circuit.

## Klíčová slova

Fyzikální simulace, fyzika, simulace, auto, kolize, závody, OpenSceneGraph, OSG, Bullet, hra

## Keywords

Physical simulation, physics, simulation, car, collision, racing, OpenSceneGraph, OSG, Bullet, game

## Citace

Javorka Marián: Fyzikální simulace v grafické scéně, diplomová práce, Brno, FIT VUT v Brně, 2011

# Fyzikální simulace v grafické scéně

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Jana Pečivy, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Marián Javorka

24.5.2011

## Poděkování

Velmi rád bych poděkoval Ing. Janovi Pečivovi, Ph.D. za odbornou pomoc a rady při tvorbě diplomové práce. Dále bych chtěl poděkovat Bc. Radovanovi Kovačevovi za pomoc při výběru a tvorbě grafických prvků výsledné aplikace a všem, kteří se podíleli na testování výsledného programu.

© Marián Javorka, 2011

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..*

# Obsah

Obsah .....	1
1 Úvod.....	2
2 Použité technológie.....	3
2.1 OpenSceneGraph.....	3
2.2 Bullet .....	3
2.2.1 Bullet Collision Detection.....	4
2.2.2 Bullet Dynamics.....	5
2.3 osgBullet.....	6
3 Teória k pohybu vozidla .....	8
3.1 Dynamika vozidla v smere akcelerácie .....	8
3.1.1 Vozidlo na rovine v pokoji .....	8
3.1.2 Výpočet polohy ťažiska .....	9
3.1.3 Akcelerujúce vozidlo na rovine .....	9
3.2 Hnacia sústava.....	10
3.2.1 Výkon motora .....	10
3.2.2 Súčasti hnacej sústavy.....	11
3.3 Riadenie.....	12
4 Implementácia.....	15
4.1 Tvorba vozidla.....	15
4.1.1 Fyzikálny model auta.....	15
4.1.2 Pripravenie vozidla na vloženie do scény .....	18
4.2 Tvorba fyzikálneho prostredia.....	20
4.3 Heads-up displej.....	21
4.4 Tvorba trate .....	23
4.4.1 Automotodrom Brno .....	23
4.4.2 Postup tvorby .....	24
4.4.3 Výjazdové zóny a obrubníky .....	27
4.5 Počasie.....	28
4.6 Vstupné menu.....	30
4.6.1 Úvodná stránka .....	30
4.6.2 Nastavenie vozidiel.....	31
4.6.3 Nastavenie trate.....	32
4.7 Ovládanie hry .....	32
4.7.1 Experimentálne ovládanie hry .....	33
4.8 Použité zdroje.....	33
4.9 Zahrievanie pneumatík.....	34
Záver.....	35

# 1 Úvod

Fyzikálna simulácia je prostriedok na zobrazenie priebehu nejakého zvoleného fyzikálneho javu, alebo ich skupiny, ktorý sa simuláciou napodobní. Výstup simulácie môže byť zobrazený graficky, alebo textovým výstupom dôležitých hodnôt v určitých časových úsekoch.

Veľké využitie má v simulovaní rôznych fyzikálnych pokusov, ktoré sú v bežných laboratórnych podmienkach ťažko realizovateľné, alebo dokonca nerealizovateľné. Jedná sa spravidla o pokusy, ktoré prebiehajú veľmi rýchlo, resp. pomaly na to aby boli sledované, alebo finančne náročné projekty ako napríklad určenie výdrže zmesi použitej pri tvorbe závodnej pneumatiky pri vysokej záťaži.

V posledných rokoch sa dostáva do popredia fyzikálna simulácia aj v počítačových a konzolových hrách. V ďalšom texte budeme pod pojmom *hra* rozumieť práve tieto typy hier. V počiatkoch moderných hier (koniec 80tych a začiatok 90tych rokov 20.storočia) mali hry veľmi jednoduchú fyzikálnu simuláciu a grafický výstup. Bolo to spôsobené hlavne nižším výpočtovým výkonom vtedajších PC. Postupným zvyšovaním výkonu PC a herných konzol sa zvyšovala úroveň výstupu a v dnešnej dobe už máme k dispozícii veľmi reálne simulátory s výbornou grafikou, niekedy skoro zhodnou s realitou, napr. F1 2010 od Codemasters, Gran Turismo 5 od Sony, alebo Forza Motorsport 3 od spoločnosti Microsoft. Avšak s narastajúcou presnosťou fyzikálneho spracovania nastáva, hlavne u závodných hier, problém, že nie každý hráč dokáže ovládať silné auto a preto sa stále produkujú aj rôzne arkádové tituly, ktoré robia kompromis medzi hrateľnosťou a vernosťou spracovania sa auta. Medzi hlavných predstaviteľov arkádového štýlu patrí séria Need for speed od Electronic Arts.

Osobne mi viac imponujú simulátory ako arkády. Človek musí stále dávať pozor napríklad na množstvo aplikovaného plynu, aby sa pri výjazde zo zákruty neroztočil a bezduché stláčanie plynu až na podlahu k ničomu nevedie. Preto som sa rozhodol pre túto prácu, ktorej cieľom je vytvoriť a následne graficky demonštrovať fyzikálnu simuláciu auta. Tvorba jednoduchého závodného simulátoru mi pripadala ako najrozumnejšie riešenie. Mojim hlavným cieľom bolo vytvorenie takého simulátoru, ktorý by umožňoval jazdiť na akomkoľvek type auta od bežných cestných vozidiel, cez terénne autá až po špičkové závodné špeciály tak, aby stačilo nastaviť správne parametre vozidla.

V druhej kapitole sú popísané tri najhlavnejšie knižnice, pomocou ktorých je práca vytvorená – OpenSceneGraph, Bullet a osgBullet. Tretia kapitola obsahuje základné teoretické poznatky o pohybe vozidla, ale aj o vozidle ako stroji. Tieto informácie sú potrebné pre pochopenie problematiky práce. Štvrtá kapitola vysvetľuje postup pri samotnej implementácii výslednej aplikácie. Obsahom tejto kapitoly je popis tvorby fyzikálneho modelu vozidla, trate a ostatných súčastí aplikácie. Ďalej je tu vypísané ovládanie hry a súhrn použitých zdrojov pri tvorbe. V záverečnej kapitole je celkové zhrnutie práce spolu s návrhmi na ďalšie vylepšovanie projektu.

Pri tvorbe semestrálneho projektu bol vytvorený fyzikálny model vozidla, ktorý bol potom už len s minimálnymi úpravami použitý v diplomovej práci. Spolu s fyzikou auta bolo v semestrálnom projekte vytvorené veľmi jednoduché fyzikálne prostredie, ktoré zahŕňalo iba gravitáciu. Taktiež toto prostredie bolo mierne rozšírené, hlavne o zohľadnenie odporu vzduchu a o dynamickú zmenu intenzity zrážok počas behu aplikácie. Tieto zmeny sú bližšie popísané v štvrtej kapitole.

## 2 Použité technológie

Celý program bol vypracovaný v programovacom jazyku C++ s použitím knižníc OpenGL, OpenSceneGraph, Bullet a osgBullet, ktorá integruje Bullet do OpenSceneGraph. OpenGL a C++ sú väčšine programátorov dobre známe, preto som sa rozhodol bližšie popísať iba knižnice OpenSceneGraph, Bullet a osgBullet.

Pri tvorbe aplikácie som navyše využil API<sup>1</sup> MySQL Connector/C++, pomocou ktorého sa aplikácia pripája na vzdialený server a umožňuje on-line uchovávanie údajov.

### 2.1 OpenSceneGraph

OpenScenegraph (ďalej len OSG) je veľmi rozšírený a obľúbený multiplatformný nástroj pre vývoj aplikácií ako napríklad vizualizácie, hry, simulátory, virtuálna realita, filmy a pod. Je to objektovo orientovaný framework vytvorený v jazyku C++ a postavený nad OpenGL. Obsahuje veľa funkcií na prácu s kamerou, načítavanie súborov a iných často používaných vecí, ktoré môže programátor využiť a nemusí sa starať o ich vytváranie. OSG obsahuje užitočné veci ako napr.:

- Načítavanie rôznych formátov 3D objektov, obrázkov, či fontov
- Podporuje rôzne druhy orezávania neviditeľných častí scény, Level Of Detail, vertex arrays, vertex buffer objekty, ...
- Samostatné knižnice na prácu s časticami, textom, tieňmi, terénom, ...

Ďalšou veľkou výhodou OSG je podpora zásuvných modulov a množstvo dostupných knižníc postavených na OSG, čo umožňuje veľmi rýchlu a pohodlnú integráciu do aplikácie. Medzi takéto knižnice patrí napríklad osgPPU, osgEphemeris, osgTDS, či nižšie popísaná osgBullet.

OsgPPU uľahčuje tvorbu post-processing efektov, osgEphemeris vytvára dynamické virtuálne prostredie pre oblohu obsahujúcu slnko, planéty, hviezdy, ktoré zhruba odpovedajú pozíciám vesmírnych telies vo zvolenom čase. OsgTDS je zasa nástroj na deformáciu terénov.



Obr. 2.1: OpenSceneGraph logo

### 2.2 Bullet

Bullet je multiplatformný open-source fyzikálny engine publikovaný pod zľib licenciou, ktorý je široko využívaný v rôznych profesionálnych aplikáciách, či filmoch. Z hier je to napríklad Grand Theft Auto IV, Madagascar Kartz, Trials HD. Simulácie pomocou Bullet môžeme vidieť vo filmoch 2012, Hancock, Bolt, Sherlock Holmes, Shrek 4 a iných. Tak isto sa využíva v 3D modelačných

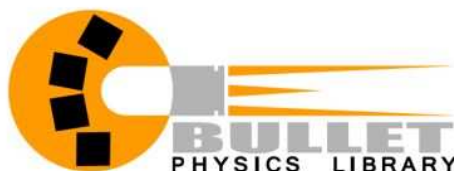
---

<sup>1</sup> Application Programming Interface – rozhranie pre programovanie aplikácií



nástrojoch (Cinema4D, LightWave, Blender) a rôznych engine-och (OGRE, Irrlicht Engine). Hlavným autorom je Erwin Coumans, zamestnanec Sony Computer Entertainment US R&D.

Bullet by sa dal rozdeliť na dve hlavné časti – Bullet Dynamics a Bullet Collision Detection, kde Collision Detection sa dá použiť ako samostatné SDK<sup>2</sup> bez Bullet Dynamics iba na výpočet kolízií. Tak isto poskytuje rôzne optimalizácie pre PlayStation3 Cell SPU, CUDA a OpenCL.



Obr. 2.2: Bullet physics library logo

## 2.2.1 Bullet Collision Detection

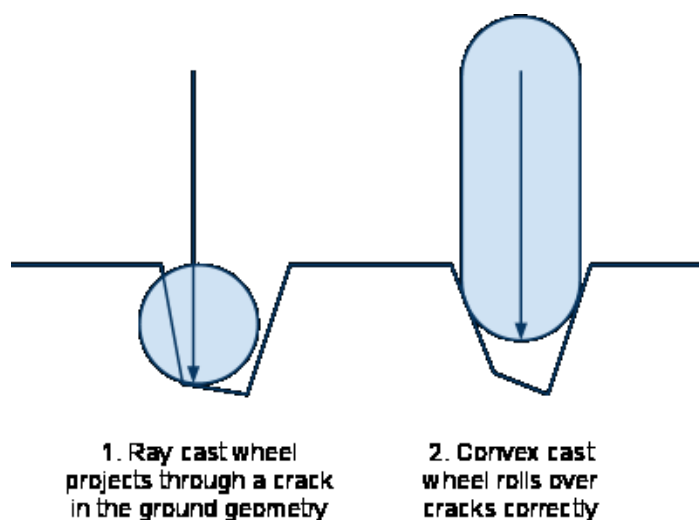
Bullet Collision Detection nám poskytuje dva typy výpočtov kolízií:

- **A posteriori** (z *lat.* spätne, dodatočne), nazývaná aj *diskrétna* kolízna detekcia. Simuláciou sa postupuje malými časovými krokmi a kontroluje sa, či niektoré objekty medzi sebou nekolidujú. V každom kroku sa vytvorí zoznam kolidujúcich objektov a ich trajektória sa upraví tak, aby odpovedala prebehnutej kolízii. Preto sa nazýva *a posteriori*, lebo zvyčajne zachytíme kolíziu až potom, ako k nej dôjde. Medzi hlavné výhody tejto metódy patrí jednoduchší výpočet kolízií, lebo sa zanedbávajú niektoré fyzikálne veličiny ako trenie, elasticita povrchu kolíznych objektov, či ich deformácie.
- **A priori** (z *lat.* vopred, napred), nazývaná aj *spojitá* kolízna detekcia. Na rozdiel od diskkrétnej detekcie sa tu prepočítava trajektória pohybu každého objektu a teda sa vie vopred určiť kedy sa dva objekty stretnú – preto sa nazýva *a priori*. Táto metóda je presnejšia ako predchádzajúca, lebo sa dajú zohľadniť rôzne fyzikálne vlastnosti materiálov, je však o jednu dimenziu – časovú – zložitejšia [15].

Pri kontrolovaní kolízií sa dá postupovať dvomi spôsobmi - *Ray casting* a *Convex casting*. Pri metóde ray casting sa vysiela z objektu lúč a kontroluje sa priesečník s iným objektom. Takto sa získa vzdialenosť od daného objektu. Na rozdiel od jednoduchšej ray casting verzie sa pri convex casting metóde berie do úvahy celé konvexné teleso. Na Obr. 2.3 je znázornený rozdiel v chovaní kolies auta pri prechádzaní nerovným terénom pri kolesách typu ray casting a convex casting [2].

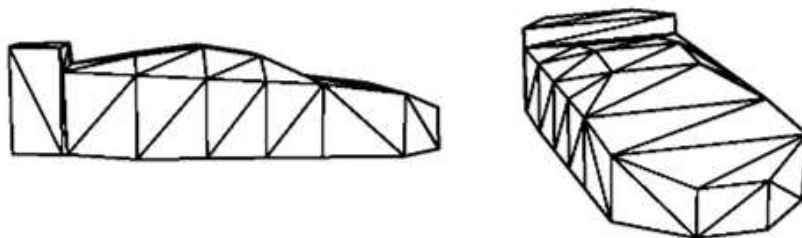
---

<sup>2</sup> Software development kit – nástroj pre vývoj aplikácií



Obr. 2.3: Porovnanie chovania Ray cast a Convex cast kolies auta

Medzi základné telesá, ktoré sú v Bullet Collision Detection podporované patria bežné 3D telesá ako kapsula, valec, kužeľ, guľa a kocka. Je možné si však vytvoriť vlastný kolízny objekt z ľubovoľnej trojuholníkovej siete (Obr. 2.4). V dokumentácii ku knižnici sa odporúča dodržať maximum 100 trojuholníkov. Sám som si ale skúšal zložitejšie telesá a pri simulácii s nimi nebol žiadny problém. Nakoniec som sa to rozhodol demonštrovať aj v mojej práci, kde používam kolízne teleso pre závodný okruh, ktoré sa skladá z 27130 trojuholníkov. Toto teleso je však brané ako statické a teda sa nepočíta s jeho pohybom. Predpokladám, že tu bude menej výpočtov ako pri dynamickom telese, preto pri tvorbe kolíznych telies vozidiel som dodržiaval odporúčaný limit 100 trojuholníkov.



Obr. 2.4: Príklad vlastného kolízneho telesa auta (80 trojuholníkov)

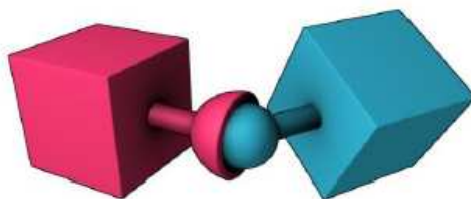
## 2.2.2 Bullet Dynamics

Bullet Dynamics nám dovoľuje operovať s dvomi typmi telies – *rigid* a *soft bodies*, ktoré majú, na rozdiel od častíc, 6 stupňov voľnosti pohybu v priestore. Môžeme teda určiť tri súradnice ťažiska ( $x, y, z$ ), tak ako pri časticiach, ale k tomu ešte rotáciu telesa okolo každej z osí.

*Rigid body* je označenie pre nedeformovateľné teleso, ktoré je charakteristické svojou hmotnosťou, telesným objemom, zotrvačnosťou atď. Použitie týchto telies dokonale postačuje pri simuláciách, kde záleží na pohybe a rotácii telies, ale deformácia telies nemá na výsledok významný vplyv. Práve tento typ kolíznych telies som sa rozhodol použiť pri implementovaní tejto práce.

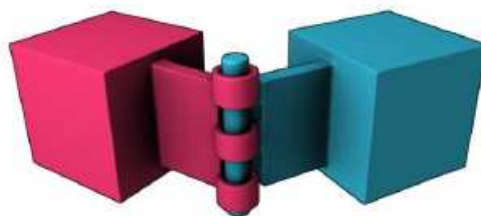
Bullet poskytuje niekoľko možností spojov dvoch telies. Medzi základné typy patria:

- kĺbový spoj (**point to point constraint**) – dovoľuje pohyb aj rotáciu jedného telesa vo všetkých 6 smeroch, tak, že je závislé na druhom pripojenom telese



Obr. 2.5: Point to point constraint

- čapový spoj (**hinge constraint**) – teleso môže iba rotovať okolo jednej osi. Využíva sa hlavne pri simulovaní dverí a okien



Obr. 2.6: Hinge constraint

- ragdoll spoj (**cone twist constraint**) – využíva sa na simulovanie pohybovej sústavy zvierat a ľudí. Je to špeciálny typ kĺbového spoja, ktorý má pridané limity v pohyboch a rotáciách.

*Soft body* je označenie pre deformovateľné teleso. Soft bodies sú implementované nad rigid bodies a využívajú sa na simulovanie telies, ktoré môžu meniť svoj tvar, ale očakáva sa od nich, že si do určitého bodu svoj tvar udržia. Teda sú vhodné na simuláciu šiat, lana, vlasov, svalov, tuku, vegetácie, nafukovacích lôpt a pod. Naopak nie sú vhodné na simuláciu tekutiny, pretože tá si svoj tvar neudrží.

Viac informácií o knižnici Bullet nájdete v literatúre [1], [3], [6] a [7].

## 2.3 osgBullet

Aby sa dalo tvoriť aplikáciu zároveň s Bulletom, aj s OSG je treba integrovať Bullet do OSG. Na to slúži knižnica osgBullet, ktorá vznikla spoluprácou Paula Martza, Ames Lab a ARDEC, ktorá je pod licenciou LGPL v2.1.



**Obr. 2.7: osgBullet logo**

OsgBullet poskytuje mimo iné:

- funkcie na konvertovanie rigid bodies z Bulletu na graf scény z OSG a naopak
- funkcie na konvertovanie maticových a vektorových dátových typov medzi Bulletom a OSG
- triedu MotionState, ktorá dovoľuje Bulletu špecifikovať transformačnú maticu v OSG

Viac informácií je dostupných na [4].

## 3 Teória k pohybu vozidla

Predmetom tejto kapitoly je vysvetlenie základných pojmov týkajúcich sa dynamiky vozidla a okrajové nahliadnutie do teórie pohybu motorového vozidla. V celej práci sa berie do úvahy symetrický štvorkolesový automobil s riadením predných kolies. Preto sa aj v tejto kapitole zameriame iba na tento typ vozidla a nebudeme uvažovať špeciálne prípady ako prívesy, 6 kolesové a viackolesové vozidlá a podobne.

Kapitola je rozdelená do troch častí, kde sú postupne prebrané dynamika vozidla v smere akcelerácie, hnacia sústava a riadenie.

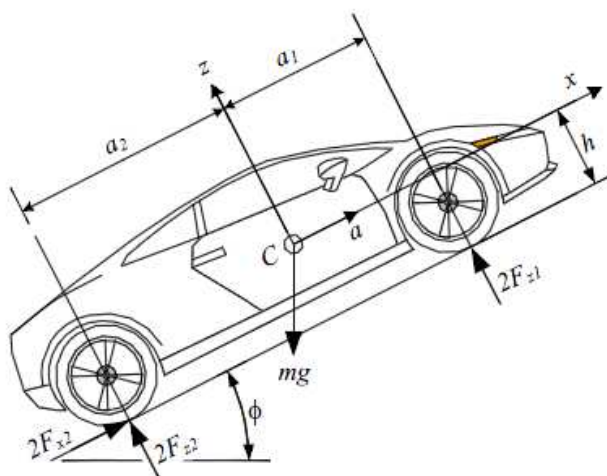
Celá kapitola, vrátane väčšiny ilustračných obrázkov vychádza z literatúry [8].

### 3.1 Dynamika vozidla v smere akcelerácie

Subjektom tejto podkapitoly je priamočiary pohyb vozidla. Budeme zisťovať ako sa mení rozloženie hmotnosti auta na nerovnom povrchu a pri akcelerácii, či decelerácii. Pri týchto výpočtoch budeme odpor vzduchu zanedbávať.

#### 3.1.1 Vozidlo na rovine v pokoji

Pre každé vozidlo platí, že jediný kontakt so zemou prebieha iba cez jeho kolesá. Preto sa celá váha auta delí medzi tieto štyri plochy. Rozdelenie váhy však neprebíha rovnomerne. Je závislé na pozícii ťažiska vozidla. Väčšina dnešných cestných vozidiel má umiestnený motor vpredú a zároveň má náhon na predné kolesá. Hmotnosť motora spôsobuje, že sa celé ťažisko posúva mierne dopredu. To je v tomto prípade prínos, lebo predné kolesá sú tým pádom viac zaťažené a to spôsobuje lepšiu priľnavosť predných kolies. U áut, ktoré majú zadný náhon je naopak snaha zvýšiť priľnavosť zadných gúm. Ale treba dbať aj na to, že predné kolesá potrebujeme na zabáčanie a preto ich nemôžeme úplne odľahčiť. Preto sa v niektorých športových autách umiestňuje motor do stredu – teda medzi kolesá. Tak isto sa používajú rôzne prítlačné krídla a aerodynamické prvky, ktoré hlavne vo vyšších rýchlostiach, tlačia vozidlo k zemi, zvyšujú priľnavosť gúm a tým hlavne jeho ovládateľnosť.



Obr. 3.1 Vozidlo zaparkované na naklonenej rovine

Na Obr. 3.1 je znázornené vozidlo s hmotnosťou  $m$  zaparkované na naklonenej rovine, ktorá je naklonená v uhle  $\phi$ . Pre tento stav platia rovnice

$$F_{z_1} = \frac{1}{2}mg \frac{a_2}{l} \cos \phi - \frac{1}{2}mg \frac{h}{l} \sin \phi \quad (3.1)$$

$$F_{z_2} = \frac{1}{2}mg \frac{a_1}{l} \cos \phi + \frac{1}{2}mg \frac{h}{l} \sin \phi \quad (3.2)$$

$$F_{x_2} = \frac{1}{2}mg \sin \phi \quad (3.3)$$

kde  $F_{z_1}, F_{z_2}$  sú sily pôsobiace na predné, resp. zadné kolesá,  $a_1$  značí vzdialenosť od ťažiska  $C$  po prednú nápravu,  $a_2$  je vzdialenosť ťažiska od zadnej nápravy.  $F_{x_2}$  označuje potrebnú brzdnú silu, aby sa vozidlo nepohybovalo smerom dolu a  $h$  je výška ťažiska. Vzdialenosť prednej a zadnej nápravy je  $l$ ,

$$l = a_1 + a_2 \quad (3.4)$$

Špeciálny prípad nastáva ak je naklonenie roviny nulové – teda je vozidlo vo vodorovnej polohe. Vtedy sa rovnice (3.1) až (3.3) zjednodušia na rovnice (3.5) až (3.7)

$$F_{z_1} = \frac{1}{2}mg \frac{a_2}{l} \quad (3.5)$$

$$F_{z_2} = \frac{1}{2}mg \frac{a_1}{l} \quad (3.6)$$

$$F_{x_2} = 0 \quad (3.7)$$

### 3.1.2 Výpočet polohy ťažiska

Hlavne u závodných automobilov je žiaduce aby bolo ťažisko vozidla čo najnižšie. To zvyšuje ovládateľnosť vozidla.

Pozdĺžna poloha ťažiska sa dá zistiť tak, že si odmeriame silu pôsobiacu na predné, alebo zadné kolesá a z rovnice (3.5), resp. (3.6) vyjadríme  $a_2$ , respektíve  $a_1$ .

Laterálna poloha je jednoznačná, keďže vychádzame z predpokladu, že je vozidlo symetrické, teda ťažisko je v tomto smere presne v strede vozidla. Ak by však tento predpoklad neplatil, tak polohu ťažiska získame odvážením jednej strany vozidla a obdobným spôsobom ako u pozdĺžnej polohy zistíme vzdialenosť ťažiska od kolies.

Výpočet výšky ťažiska je náročnejší, jeho postup je v literatúre [8] na str.42.

### 3.1.3 Akcelerujúce vozidlo na rovine

V podkapitole 3.1.1 sme vypočítali sily pôsobiace na kolesá, keď je vozidlo v pokoji. Pravá časť rovníc (3.1), (3.2), (3.5) a (3.6) sa nazýva statická časť. Ak automobil akceleruje, tak musíme rovnice

rozšíriť o tzv. dynamickú časť, ktorá je závislá práve na veľkosti akcelerácie  $a$ . Týmto rozšírením teda dostaneme všeobecný tvar

$$F_{z_1} = \frac{1}{2}mg \frac{a_2}{l} \cos \phi - \frac{1}{2}mg \frac{h}{l} \sin \phi - \frac{1}{2}ma \frac{h}{l} \quad (3.8)$$

$$F_{z_2} = \frac{1}{2}mg \frac{a_1}{l} \cos \phi + \frac{1}{2}mg \frac{h}{l} \sin \phi + \frac{1}{2}ma \frac{h}{l} \quad (3.9)$$

a pre akceleráciu na vodorovnej ploche potom

$$F_{z_1} = \frac{1}{2}mg \frac{a_2}{l} - \frac{1}{2}ma \frac{h}{l} \quad (3.10)$$

$$F_{z_2} = \frac{1}{2}mg \frac{a_1}{l} + \frac{1}{2}ma \frac{h}{l} \quad (3.11)$$

Ak pre akceleráciu platí  $a > 0$ , tak automobil zrýchľuje. Naopak ak je v platnosti vzťah  $a < 0$ , jedná sa o spomaľovanie.

## 3.2 Hnacia sústava

Maximálna možná akcelerácia každého vozidla závisí na dvoch faktoroch. Prvým je maximálny krútiaci moment na poháňaných kolesách, čo závisí hlavne na výkone motora a prevodovky. Práve výkon motora a prevodovky sú obsahom tejto podkapitoly.

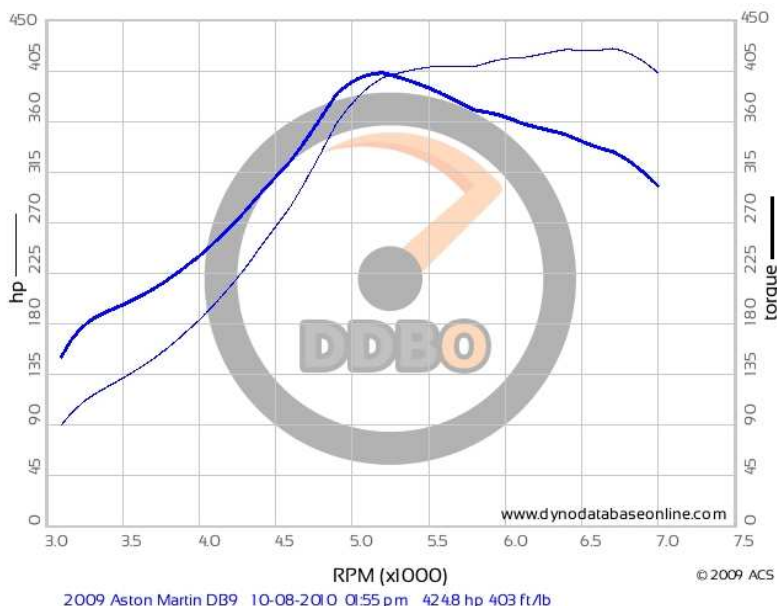
Tým druhým faktorom, vplývajúcim na akceleráciu automobilu, je maximálna trecia sila na kontaktoch kolies s vozovkou. Tá závisí od veľkosti vznikajúceho trenia medzi gumou kolies a povrchom pod nimi. Veľkosť trenia závisí od mnohých ďalších faktorov, ako zloženie gumy, či vlastnosti povrchu. Na suchom asfalte je vyššia priľnavosť ako na mokrom. Na pórovitom betóne je zasa vyššia priľnavosť ako na hladkom asfalte. Naopak na špeciálnom asfalte, ktorý sa používa na závodných okružoch, sa aj vďaka drobným časticiam gumy dosahuje ešte vyššie trenie. Nemalý vplyv má aj teplota, či hustota tlaku v pneumatikách. Prehustené, alebo málo nafúkané pneumatiky majú menšiu dotykovú plochu ako tie so správnym tlakom. Problematika priľnavosti a simulácia gúm je tak obširná a náročná, že by mohla byť predmetom samostatnej diplomovej, či dizertačnej práce.

### 3.2.1 Výkon motora

Výkon motora sa meria pomocou silomera (*ang. dynamometer*) a meria sa sila a krútiaci moment motora. Výsledok merania je zväčša zobrazený v grafe, ktorý je vidno na Obr. 3.2., kde sú znázornené krivky, určujúce krútiaci moment (torque) a silu motora v koňoch (hp) v závislosti od aktuálnych otáčok motora.

Takýto graf sa v angličtine nazýva *dyno graph* – odvodené od meracieho prístroja. Na grafe je vidno, že pri nižších otáčkach je vyšší krútiaci moment a až pri vyšších otáčkach ho sila prevyšuje. Preto ak majú dve autá rovnakú silu motora (hovorovo povedané „rovnaký počet koní“), ale jedno má vyšší krútiaci moment, tak toto bude v nižších otáčkach svižnejšie ako to druhé. Z grafu sa dá vyčítať aj správny postup radenia pre dosiahnutie maximálneho výkonu. Dôvodom zaraďovania vyššieho prevodového stupňa je zníženie otáčok motora. Pri pretekaní by sa malo zaraďiť tak, aby po zaraďení

nového stupňa neklesli otáčky pod špičku krivky krútiaceho momentu, teda na Obr. 3.2 by po zaradení nemali klesnúť otáčky pod zhruba 5200 otáčok za minútu. Z grafu sa dajú vyčítať aj maximálne a minimálne otáčky, v ktorých motor pracuje. Pre znázornený Aston Martin je minimum, tzv. voľnobeh, v zhruba 3100 otáčkach za minútu a maximum pri 7000, kedy sa auto dostáva do červeného pásma na otáčkomere. Bežné osobné vozidlá však majú oveľa nižší voľnobeh, zhruba na 1200 – 1700 otáčkach.



Obr. 3.2: Charakteristika motora vozidla Aston Martin DB9<sup>3</sup>

### 3.2.2 Súčasti hnacej sústavy

Motor nám teda vytvorí nejaký krútiaci moment. Ten je však treba preniesť na kolesá. Na to slúžia rôzne komponenty, ktoré sú zapojené do *hnacej súpravy*. Na Obr. 3.3 je znázornené zapojenie komponentov v aute so zadným náhonom. V našom prípade nám však postačí, ak budeme rátať so spojením týchto komponentov: motor (engine), prevodovka (gearbox), diferenciál (differential), koleso (wheel).

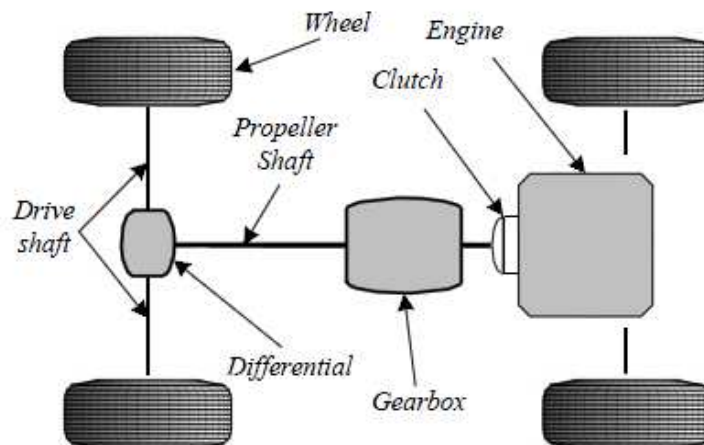
Prevodovka slúži na násobenie krútiaceho momentu motora podľa zaradeného prevodového stupňa. Každý prevodový stupeň je charakteristický svojím pomerom. Tento pomer sa získa vydelením počtu zubov ozubených kolies zapojených do mechanizmu. Prevodové stupne s vyšším pomerom dávajú väčšiu silu – teda zrýchlenie, ale menšiu finálnu rýchlosť. Naopak zmenšením pomerov sa znižuje sila a zvyšuje rýchlosť. V tabuľke Tab. 3.1 je príklad ako by mohli byť nastavené prevodové stupne na aute Aston Martin DB9.

prevodový stupeň	1	2	3	4	5
pomer	3.723:1	2.669:1	1.680:1	1.151:1	0.744:1

Tab. 3.1: Pomery prevodových stupňov auta Aston Martin DB9

<sup>3</sup> <http://www.dynodatabaseonline.com/dyno/index.php?runid1=117&runid2=0&hp=1&torque=1&gb=0>





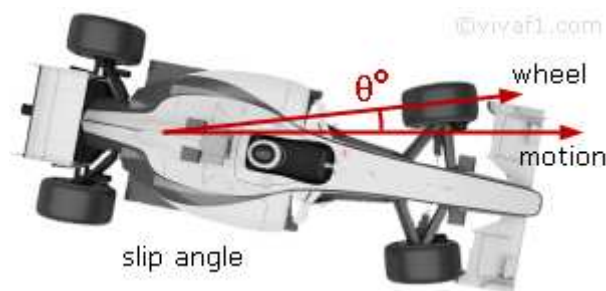
Obr. 3.3: Zapojenie komponentov v aute so zadným náhonom kolies

Prevodový stupeň 1 má pomer 3.723:1, čo znamená, že motor urobí 3.723krát viac otáčok ako hnací hriadeľ (propeller shaft), ktorý je na výstupe prevodovky. Naopak posledný prevodový stupeň má pomer 0.744:1 z čoho dostávame, že motor urobí menej otáčok ako hnací hriadeľ. Tento prevodový stupeň sa nazýva *overdrive* (rýchlobeh)[9] a slúži na jazdenie autom pri konštantnej rýchlosti s menšou rýchlosťou motora a teda zníženou spotrebou paliva. U bežných 5stupňových, resp. 6stupňových prevodoviek osobných áut sú prvé tri stupne s pomerom vyšším ako 1:1. Štvrtý prevodový stupeň má spravidla pomer presne 1:1, čiže motor má také iste otáčky ako hnací hriadeľ. Piaty, resp. piaty a šiesty prevodový stupeň sú potom rýchlobehy.

Diferenciál má za úlohu rozdeliť vstupný, už sprevodovaný, krútiaci moment na kolesá. Diferenciál je tiež zložený z ozubených kolies a teda má tiež svoj pomer, ktorý spravidla býva okolo 3:1 až 4:1. Ak sú obe kolesá na asfalte (majú rovnaký odpor) a vozidlo ide rovno, tak diferenciál rozdelí krútiaci moment rovnomerne na obe kolesá. Keď vozidlo začne zabáčať, tak vnútorné koleso získava väčší odpor a diferenciál prinúti vonkajšie koleso zvýšiť otáčky tým, že odoberie otáčky vnútornému kolesu a pridá ich vonkajšiemu. Týmto sa zabráni prejazdu zákrutou v šmyku, šetria sa pneumatiky a zvyšuje sa stabilita auta v zákrute. Nevýhoda je však v tom, že ak je rozdiel odporov kolies veľmi vysoký, napríklad jedno koleso je na asfalte a jedno na ľade, tak koleso s vyšším odporom sa zavrie úplne a druhé koleso dostáva dvojnásobné otáčky. Zvýšenie otáčok pre koleso na ľade však ničomu nepomôže a hlavne pri rozjazde dochádza k problémom. Tento problém sa rieši napríklad uzávierkou diferenciálu, ktorá zabráni úplnému zablokovaniu kolesa s vysokým odporom. Pre aktuálne potreby aplikácie nám stačí poznať pomer diferenciálu. Riešenie ostatných činností by mohlo byť predmetom ďalšieho vývoja.

### 3.3 Riadenie

Každá pneumatika má svojimi vlastnosťami udanú maximálnu postrannú silu, ktorú dokáže vyvinúť. Pri riadení vozidla v nižších rýchlostiach je pre zabočenie potrebná menšia postranná sila, naopak pri vyšších rýchlostiach je treba väčšia. Hlavne pri vyšších rýchlostiach pneumatiky nedokážu dostatočnú silu vyvinúť a vtedy vzniká *uhol sklzu* (angl. *slip angle* alebo aj *sideslip angle*), ktorý udáva rozdiel medzi natočením kolies a skutočným smerom vozidla. Tento uhol je vidno na Obr. 3.4

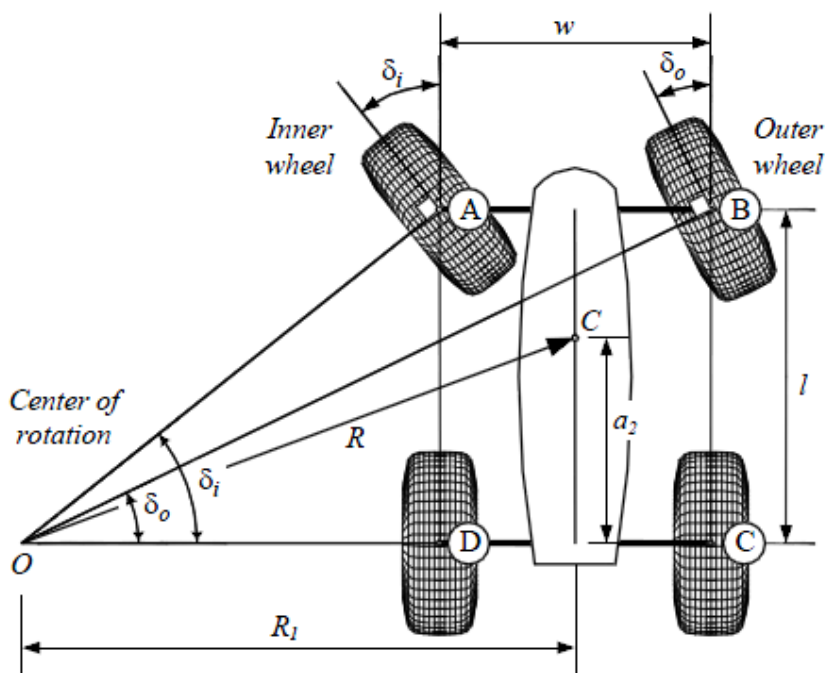


Obr. 3.4 Uhol sklzu<sup>4</sup>

Pri riadení vozidla vo veľmi nízkej rýchlosti dochádza pri zabáčaní ku stavu medzi vnútornými a vonkajšími kolesami, ktorý umožňuje autu zabáčať tak, aby bol uhol sklzu nulový. Tento stav je zobrazený na Obr. 3.5 a nazýva sa *Ackermanov stav*. Platí preň rovnica

$$\cot \delta_o - \cot \delta_i = \frac{w}{l} \quad (3.12)$$

kde  $\delta_o, \delta_i$  sú uhly vonkajšieho a vnútorného kolesa,  $w$  je vzdialenosť medzi prednými kolesami.



Obr. 3.5 Vozidlo zabáčajúce doľava

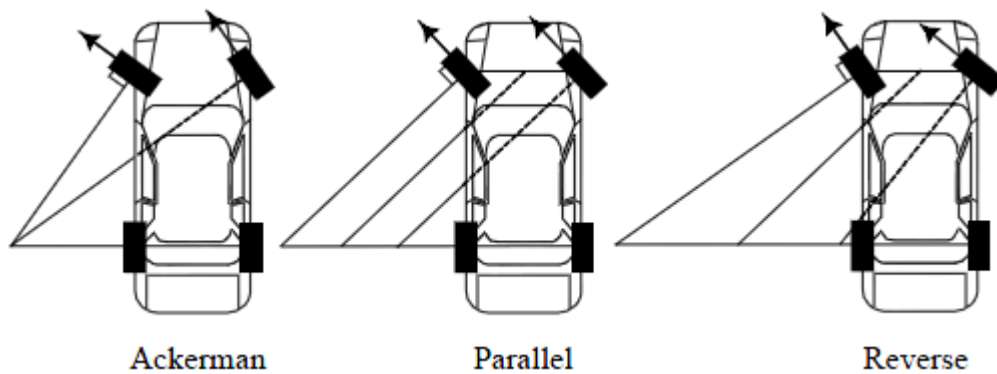
Pri uplatnení Ackermanovho stavu platí, že uhol vnútorného kolesa je väčší ako uhol vonkajšieho. Riadiace mechanizmy, ktoré toto umožňujú sa nazývajú Ackermanove mechanizmy. Zvyšovaním rýchlosti vozidla sa však zvyšuje uhol sklzu. Zároveň sa váha prenáša na vonkajšie kolesá, ktoré znášajú oveľa vyššiu záťaž ako vnútorné. Zvyšovaním záťaže kolesa sa znižuje uhol

<sup>4</sup> [http://www.vivaf1.com/glossary\\_s.php](http://www.vivaf1.com/glossary_s.php)

sklzu, ktorý je potrebný na dosiahnutie maximálnej postrannej sily. A keďže majú vnútorné kolesá napojené na Ackermanove riadiace mechanizmy väčší uhol, znamená to, že pracujú pri vyššom uhle sklzu ako je potrebný na dosiahnutie maximálnej laterálnej sily. Preto pri zabáčaní vo vyšších rýchlostiach treba zmenšiť uhol otočenia vnútorného kolesa.

U závodných áut sa preto väčšinou používa paralelný riadiaci mechanizmus, alebo obrátený Ackermanov mechanizmus. Tieto vozidlá však majú problém pri zabáčaní v nižších rýchlostiach.

Na Obr. 3.6 sú zobrazené všetky spomenuté typy riadiacich mechanizmov.



**Obr. 3.6 Tri typy riadiacich mechanizmov**

## 4 Implementácia

Cieľom práce bolo naštudovať si knižnice uvedené v druhej kapitole a pomocou nadobudnutých znalostí vytvoriť grafickú scénu, ktorá by zobrazovala simuláciu auta vrátane kolízií s okolitými objektmi a ostatnými autami. Pre demonštráciu vytvorenej fyzikálnej simulácie som sa rozhodol pre tvorbu jednoduchej závodnej hry. V súčasnom stave hra umožňuje hrať jednému hráčovi, alebo dvom hráčom cez vertikálne rozdelenú obrazovku. V hre je zabudovaný jednoduchý model počasia, ktorý priamo ovplyvňuje vlastnosti vozidla.

V tejto kapitole sa budú objavovať názvy použitých funkcií, tried a malé útržky zdrojových kódov, tieto budú písané strojovým písmom.

### 4.1 Tvorba vozidla

Mojim hlavným cieľom bolo vytvorenie spomínaného univerzálneho fyzikálneho modelu auta. Pre tento účel som si vytvoril triedu `Vehicle`, ktorá spája funkcie `Bulletu` aj `OSG` do jedného celku a umožňuje veľmi jednoducho pridať auto do scény.

Trieda `Vehicle` obsahuje okrem nižšie popísaného fyzikálneho modelu aj nástroje na ľahkú integráciu vozidla do grafu scény `OSG`, ktorá spočíva v týchto jednoduchých krokoch:

- Zadefinovanie ovládacích kláves vozidla
- Načítanie 3D modelu a kolízneho modelu auta a kolies
- Načítanie konfiguračného súboru auta
- Vloženie do grafu scény

Zdrojový kód vloženia auta do scény je uvedený v prílohe 1.

#### 4.1.1 Fyzikálny model auta

Knižnica `Bullet` poskytuje vývojárom aplikácií veľmi zjednodušený model auta v triede `btRaycastVehicle`. Tento model vozidla zjednodušuje simuláciu tým, že nesimuluje kolesá oddelene od karosérie spojené nejakým spojom, ale zabaľuje všetko do jedného telesa – rigid body. Výšku kolies, teda stlačenie odpruženia, počíta pomocou metódy ray casting (Obr. 2.3). Táto metóda je rýchla a vo väčšine aplikácií úplne postačujúca. Problémy by mohli nastať pri veľmi pomalom prechádzaní veľmi členitého terénu, napríklad simulácia bagrov. Vtedy by sa mohli kolesá viditeľne ponárať pod povrch vozovky.

Ja som sa rozhodol začať práve s týmto modelom. `Bullet` demonštruje príklad použitia v ukážke `VehicleDemo`. Po implementovaní tohto modelu do projektu som testovaním zisťoval čo všetko ponúka, aké sú výhody a nevýhody.

Veľkým prínosom bol podvozok spolu s odpružením. Rôznym nastavením parametrov sa dá nastaviť tuhosť odpruženia, výška pružín, podvozku a iné (výpis hlavných parametrov je uvedený v [2]). Sám som si vyskúšal nastavenia pre športové autá i pre terénne. Niekedy však treba odbočiť od reálnych hodnôt a upraviť ich tak, aby výsledné správanie auta pripomínalo reálne. Je to spôsobené tým, že celé auto je brané ako jedno teleso s ťažiskom v strede, ale v skutočnosti má auto ťažisko v rovine podvozku a motora, kde je sústredená najväčšia váha.

Ray cast model veľmi pekne reaguje na členitosť terénu a auto sa reálne správa aj pri skokoch (Obr. 4.1). Simulácia odpruženia modelu tak isto dobre reaguje na dynamickú zmenu váhu auta, teda pri akcelerácii sa dvíha predok, pri brzdení sa naopak dvíha zadok a v zákrutách sa odľahčuje vnútorná strana. Samozrejme záleží na nastavení odpruženia, pretože pri tuhom nastavení športových áut je to skoro nepostrehnuteľné.



**Obr. 4.1: Priebeh skoku z rampy pri rýchlosti 60 km/h**

Čo však v tomto modeli úplne chýba je motor. Ten je nahradený konštantnou silou aplikovanou na kolesá, ktorá dovoľovala neustále konštantné zrýchlenie auta. To je prvá vec, ktorú som do môjho fyzikálneho modelu pridal spolu s prevodovkou, keďže spolu úzko súvisia.

Najprv bolo potrebné získať charakteristiku motora, ktorá sa dá pre bežné motory nájsť bez problémov na internete. Táto charakteristika odpovedá výkonostnému grafu uvedeného v tretej kapitole na Obr. 3.2. V aplikácii sa pri vkladaní auta do scény načítajú dáta sily a krútiaceho momentu pri pár otáčkach tak, aby bol zhruba zachovaný priebeh kriviek a v simulácii sa ostatné hodnoty dopočítavajú kosínusovou interpoláciou. V tretej kapitole je tak isto uvedené, že pre základnú simuláciu motora potrebujeme okrem výkonu motora ešte pomery prevodových stupňov a diferenciálu. Tieto informácie sú tak isto voľne dostupné na internete.

Ak teda poznáme charakteristiku motora a všetky potrebné pomery prevodovky a diferenciálu môžeme podľa nasledujúcej rovnice vypočítať výslednú silu motora

$$EngineForce = throttle * Torque * DiffRatio * GearRatio * effectivity \quad (4.1)$$

kde:

- *EngineForce* je výsledná sila motora
- *throttle* je poloha plynového pedála z intervalu  $< 0,1 >$

- *Torque* je krútiaci moment v aktuálnych otáčkach motora
- *DiffRatio* a *GearRatio* sú pomery diferenciálu a zaradeného prevodového stupňa
- *effectivity* simuluje stratu výkonu pri mechanických činnostiach zariadení, zatiaľ je nastavená efektívnosť na 90%

Jedinou premennou, ktorú treba dopočítať sú aktuálne otáčky motora, aby sme vedeli získať aktuálny krútiaci moment motora. Tie vypočítame z rovnice

$$EngineRPM = \left( \frac{|speed|}{2\pi * R} \right) * DiffRatio * GearRatio \quad (4.2)$$

kde:

- *EngineRPM* je počet otáčok motora za minútu
- *speed* je rýchlosť vozidla v  $m.s^{-1}$
- *R* je polomer poháňaného kolesa v metroch

Ďalším problémom, ktorý bolo treba riešiť, je brzdný účinok motora. Bez neho by totižto pri pustení plynu auto nespomaľovalo, ale udržiavalo si konštantnú rýchlosť (ak pomineme odpor vzduchu, vozovky a pod). Presný výpočet brzdného účinku motora je veľmi náročný a silno závisí od konštrukcie motora, preto som sa rozhodol simulovať túto silu tak, že na kolesá aplikujem záporný aktuálny krútiaci moment pri polovične stlačenom plynovom pedáli. Rovnica použitá pre výpočet veľkosti brzdného účinku motora je teda

$$EngineBrakingForce = -0.5 * Torque * DiffRatio * GearRatio * effectivity \quad (4.3)$$

Celková výsledná sila je potom súčet *EngineForce* a *EngineBrakingForce*, kde platí, že vždy je aspoň jedna z nich nulová.

V aplikácii je momentálne simulovaný najjednoduchší typ diferenciálu – úplne uzavretý. Tento typ diferenciálu nedovoľuje meniť pomery medzi vnútorným a vonkajším kolesom v zatáčke. Preto je vypočítaná výsledná sila motora vždy aplikovaná rovnomerne na pravé aj ľavé poháňané koleso. Pre automobily, ktoré majú pohon na všetky štyri kolesá sa aplikuje na každé koleso štvrtina sily, pre ostatné polovica.

Pri implementovaní riadiaceho mechanizmu som sa rozhodoval, ktorý zo spomenutých mechanizmov (Obr. 3.6) použijem. Nakoniec som sa rozhodol implementovať všetky tri mechanizmy spolu s novým hybridným systémom, ktorý podľa aktuálnej rýchlosti prepína medzi dvomi mechanizmami. Pri nízkych rýchlostiach sa používa Ackermanov mechanizmus a pri vyšších sa prepne na reverzný Ackermanov systém. Primárne je zvolený Ackermanov mechanizmus, ale voľba riadiaceho systému sa dá zmeniť kedykoľvek počas hry klávesmi F1-F4. Pri prepočte uhlov kolies vychádzam z rovnice (3.12). Z nej sa dá vyjadriť

$$\delta_o = \cot^{-1} \left( \frac{w}{l} + \cot \delta_i \right) \quad (4.4)$$

pre výpočet zmenšeného uhla *vonkajšieho* kolesa pri Ackermanovom mechanizme a

$$\delta_i = \cot^{-1} \left( \cot \delta_o - \frac{w}{l} \right) \quad (4.5)$$



pre výpočet zmenšeného uhla vnútorného kolesa pri reverznom Ackermanovom mechanizme.

Poslednou implementovanou časťou bolo umožnenie voľby medzi manuálnou a automatickou prevodovkou. Automatická prevodovka je nastavená pre potreby rýchlej jazdy, teda na vyšší stupeň sa radí pri otáčkach s najvyšším výkonom motora a podraduje sa tak, aby výkon klesol čo najmenej.

### 4.1.2 Pripravenie vozidla na vloženie do scény

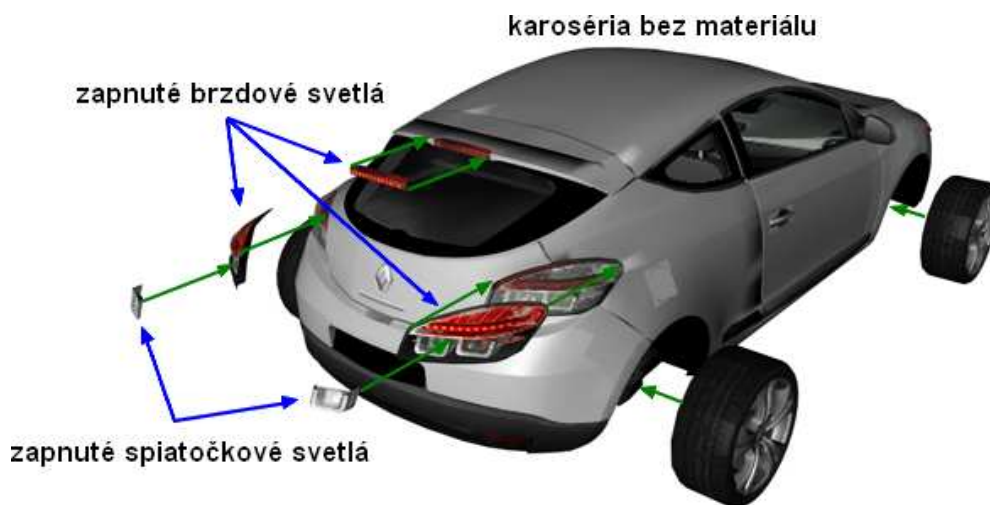
V tejto podkapitole obalíme vytvorený fyzikálny model vizuálnym 3D modelom auta a nastavíme konkrétne vlastnosti, ktoré dotvoria vozidlo podľa reálnej predlohy. V práci som sa pre demonštračné dôvody rozhodol použiť dva rôzne typy automobilov. Prvý zvolený typ je klasické rodinné auto Renault Mégane tretej generácie (Obr. 4.2 vpravo). Špecifikácie sú nastavované podľa najvýkonnejšej dostupnej verzie. Druhým autom je športový automobil Ferrari F430 (Obr. 4.2 vľavo). Takto rozdielne vozidlá som zvolil hlavne preto, aby si mohol užívateľ skúsiť fyzikálny model na rôznych typoch áut.



Obr. 4.2 Ferrari F430 a Renault Mégane III

3D modely oboch automobilov som získal zo stránky <http://www.gfx-3d-model.com>, kde sú uvedené zdarma pre nekomerčné využitie. Oba modely som následne upravil pre potreby aplikácie v modelačnom nástroji Cinema4D CE 6 od spoločnosti Maxon. Tento nástroj je tak isto dostupný zdarma.

Pre použitie v simulátore som musel odstrániť materiály laku z karosérie a rozpojiť kolesá od modelu aby sa s nimi mohlo samostatne pohybovať. Ďalej bolo treba dotvoriť brzdové svetlo a svetlo spiatocky. Zloženie výsledného modelu je zobrazené na Obr. 4.3. Podľa obrysov modelu som následne vytvoril kolízny objekt automobilu, ktorý je podobný tomu na Obr. 2.4.



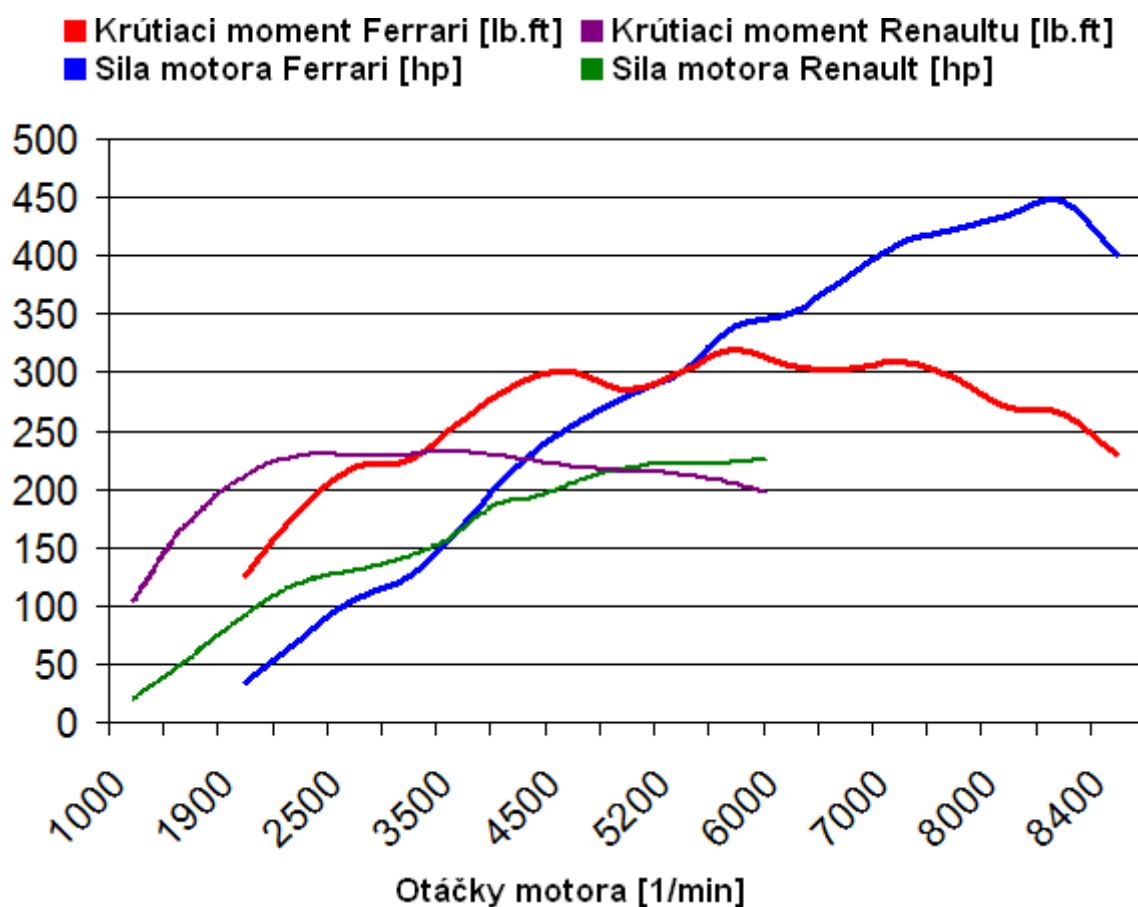
Obr. 4.3 3D model pripravený na vloženie do scény

Konfiguračný súbor, ktorý uchováva všetky potrebné informácie o vlastnostiach vozidla obsahuje viacero hodnôt (viď príloha 3). Vytiahnutím niektorých hlavných parametrov z konfiguračných súborov obidvoch automobilov a ich následným porovnaním si môžeme vytvoriť obraz o rozdielnosti týchto áut. Hlavný rozdiel je pochopiteľne v motoroch. Ich porovnanie vo forme grafu je na Obr. 4.4. Na prvý pohľad je badateľný rozdiel v rozsahu otáčok, v ktorých každý motor pracuje. Zároveň nám graf poukazuje na to, že Ferrari má zhruba 2krát silnejší motor ako Renault a poskytuje približne o 109Nm krútiaceho momentu<sup>5</sup> viac, čo je asi tretina maxima Renaultu. Ďalší dôležitý rozdiel je v pomeroch diferenciálu a rýchlostných stupňoch (Tab. 4.1).

automobil	1. stupeň	2. stupeň	3. stupeň	4. stupeň	5. stupeň	6. stupeň	diferenciál
<b>Ferrari</b>	3.29	2.16	1.61	1.27	1.03	0.82	4.3
<b>Renault</b>	3.08	1.95	1.39	1.03	0.82	0.67	3.94

**Tab. 4.1 Porovnanie pomerov prevodových stupňov a diferenciálu Ferrari a Renault**

Ak si prevedieme výkon motora cez prevodovku a diferenciál zistíme, že výkonnostný rozdiel na poháňaných nápravách ešte narastie. Na zadné kolesá Ferrari sa aplikuje pri prvom prevode stupni maximálny krútiaci moment zhruba 6128Nm. Ale predné kolesá Renaultu dostanú maximálne približne 3883Nm, čo je skoro o polovicu menej.



**Obr. 4.4 Porovnanie charakteristík motorov Ferrari a Renault**

<sup>5</sup> 1 Nm = 0,7375621 lb.ft



## 4.2 Tvorba fyzikálneho prostredia

Celé nastavenie fyzikálneho sveta umožňujú nástroje v triede `Physics`. Táto trieda ďalej umožňuje vytvoriť rigid body z akéhokoľvek kolízneho telesa spolu s možnosťou vytvoriť vlastné kolízne teleso zo siete trojuholníkov. Sieť trojuholníkov musí byť uložená v textovom súbore a musí mať nasledujúci tvar:

```
points{
  -1.0 -1.0 -1.0
  -1.0 1.0 -1.0
  1.0 -1.0 -1.0
  1.0 1.0 -1.0
  1.0 -1.0 1.0
  1.0 1.0 1.0
  -1.0 -1.0 1.0
  -1.0 1.0 1.0
}
triangles{
  0 1 3
  2 3 5
  4 5 7
  6 7 1
  1 7 5
  6 0 2
  0 3 2
  2 5 4
  4 7 6
  6 1 0
  1 5 3
  6 2 4
}
```

Tento zápis vygeneruje kolízne teleso v tvare kocky o veľkosti 2x2x2 metre. Štruktúra textového súboru vychádza zo štruktúry súborov Direct3D[12]. V prvom bloku sú súradnice ôsmich vrcholov kocky. V druhom bloku je na každom riadku trojica indexov bodov z prvého bloku, ktoré reprezentujú jeden trojuholník.

V tejto triede sú zároveň uložené aj informácie o svete, ktoré sú pre fyzikálnu simuláciu podstatné. V aktuálnom štádiu aplikácie uchováva informácie o gravitácii spolu s tlakom a teplotou vzduchu, ktorá môže byť upravovaná kedykoľvek počas behu aplikácie, napríklad v súvislosti so zmenou počasia. Posledné dve hodnoty sa používajú pre výpočet hustoty vzduchu podľa rovnice [16]

$$\rho = \frac{p}{T * R_{spec}} \quad (4.6)$$

kde

- $\rho$  značí hustotu vzduchu
- $p$  je aktuálny tlak vzduchu v Pa
- $T$  je hodnota absolútnej teploty vzduchu v K
- $R_{spec}$  je plynová konštanta – pre suchý vzduch je 287,058 J/(kg.K)

Pomocou získanej hustoty vzduchu sa pre zvýšenie reálnosti simulácie vypočítava odpor vzduchu, ktorý je aplikovaný na idúce vozidlo. Vzduch pôsobí v proti smere vozidla a hlavne pri vyšších rýchlostiach je nezanedbateľný. Priemerné vozidlo pri rýchlosti 50km/h musí prekonávať

odpor vzduchu približne 130N. Pri 100km/h veľkosť odporu narastie na 510N a pri 200km/h je to už zhruba 2050N. Ak zoberieme do úvahy rýchlosť 320km/h, čo je udávaná maximálna rýchlosť pre Ferrari F430, tak vzduch brzdí automobil silou úctyhodných 5230N. Naopak sila normálneho vetra je v tomto prípade veľmi malá a preto sa pri výpočtoch zanedbáva a vietor zostáva iba ako vizuálny doplnok. Aktuálna veľkosť sily odporu vzduchu sa dá získať pomocou vzťahu [10]

$$DragForce = \frac{1}{2} \rho * S * C_d * speed * |speed| \quad (4.7)$$

kde

- $DragForce$  je výsledná sila odporu vzduchu
- $S$  je obsah nárazovej časti auta (predok auta)
- $C_d$  označuje koeficient odporu áut (u bežných áut od 0.3 po 0.45)

V prílohe 2 sú uvedené zdrojové kódy ako pomocou Bulletu inicializovať fyzikálny svet a ako sa vytvorí rigid body.

## 4.3 Heads-up displej

Aby užívateľ mohol so simulátorom narábať, je treba mu neustále poskytovať všetky potrebné informácie o vozidle, ale aj o okolí. Osvedčený spôsob podávania informácií, nie len v závodných hrách, je tzv. *heads-up displej* skrátené *HUD*. HUD je akýkoľvek priehľadný displej, ktorý sa nachádza v zornom poli pozorovateľa a ten tak má všetky informácie priamo pred sebou.



Obr. 4.5 HUD v režime jedného hráča

V aplikácii sú použité dva Heads-up displeje – pre jedného a pre dvoch hráčov. Funkcionalitou sú rovnaké, líšia sa iba v umiestnení jednotlivých ukazovateľov. Obr. 4.5 názorne popisuje rozloženie jednotlivých komponent displeja. V ľavom hornom rohu sú údaje o počte odjazdených kôl, záznamy časov najlepšieho osobného a posledného zajazdeného kola. Nesmie chýbať ani údaj o rekorde trate. Tento údaj je však dostupný až po prihlásení do on-line režimu.

Všetky časti displeja sa plne prispôbujú veľkosti rozlíšenia obrazovky, takže pri menších monitoroch nedochádza k prílišnému zakrývaniu obrazu displejom. Samozrejme toto prispôbovanie nie je neobmedzené. Zväčšovanie prebieha do rozlíšenia 1920 pixelov na šírku. Pri väčšom rozlíšení už zostáva toto maximálne zväčšenie. Spodná hranica je nastavená na 960 pixelov na šírku. U väčších zmenšení už nebol displej čitateľný.

Po zapojení druhého hráča do hry sa otáčkomer s tachometrom zmenší aby nezaberali toľko miesta. Budíky pre druhého hráča sa zobrazia v ľavom dolnom rohu a mapka sa presunie do stredu obrazovky, kde môže slúžiť pre obidvoch hráčov naraz. Ostatné súčasti heads-up displeja zostávajú nezmenené.



Obr. 4.6 HUD v režime dvoch hráčov

Ďalšími prvkami, ktoré pomáhajú užívateľovi získať informácie o aktuálnom dianí sú pohybové rozostrenie, či dynamická kamera. Kamera je robená tak, že pri pomalých rýchlostiach je vzdialenosť od hráčovho vozidla taká malá, až je detailne vidno celé vozidlo. So zvyšujúcou sa rýchlosťou kamera zaostáva ďalej za vozidlom. Tento efekt spolu s pohybovým rozostrením navodí pocit rýchlosti. Najviac badateľné je to hlavne pri akcelerácii a brzdení, kde sa prudko mení rýchlosť vozidla a teda aj vzdialenosť kamery (Obr. 4.7).

O aktuálnom stave vozovky vizuálne informuje počasie, ktoré je podrobnejšie popísané v kapitole 4.5.



Obr. 4.7 Vzdialenosť kamery v rôznych rýchlostiach (vľavo 0km/h - vpravo 220km/h)

## 4.4 Tvorba trate

Po dokončení predchádzajúcich častí bolo treba vytvoriť priestor, kde by sa výsledný simulátor mal odohrávať. Variácií na výber bolo mnoho. Vyberal som medzi voľným priestranstvom, hlinenou cestou podobnou tým v rally a asfaltovou závodnou traťou. Kvôli závodnému duchu aplikácie odpadla možnosť voľného priestranstva a nakoniec zvíťazila voľba asfaltu.

Tvar trate je robený podľa celosvetovo známeho Masarykovho okruhu v Brne, na ktorom som sa bol v čase tvorby aplikácie aj sám pozrieť.

### 4.4.1 Automotodrom Brno

Tento 5403,19m dlhý okruh je preslávený hlavne vďaka účasti na šampionáte motoriek triedy MotoGP a Superbike. Jazdia sa tu však aj rôzne preteky cestovných vozov. Okruh obsahuje šesť ľavotočivých, osem pravotočivých zákrut a celkovo trinásť roviniek v rozmedzí dĺžky 35 až 636m. Na konci tej najdlhšej rovinky sa dosahujú rýchlosti skoro 300km/h. Brnenský okruh je zaujímavý aj výškovou členitosťou (Obr. 4.8). Je umiestnený v nadmorskej výške 450m a výškový rozdiel na trati je skoro 74m. Okruh ponúka klesanie maximálne 5% na dĺžke 410m a maximálne stúpanie je 7,5% až na 917m dlhom úseku trate. Takéto stúpanie dáva zabráť nejednému slabšiemu vozidlu.

Cesta na okruhu je široká až 15m, čo dáva priestor na mnohé predbiehacie manévry. Nevýhodou takto širokého okruhu je ale to, že skúsený jazdec v slabšom aute dokáže prejsť skoro celý okruh s plynom na podlahe. Slabším autom sa myslí obyčajné cestné vozidlo, v ktorom sa môže ktorýkoľvek vodič previezť na okruhu v stanovených dňoch pre verejnosť. Toto však neplatí o rýchlych a silných športových vozidlách, na ktorých tu jazdia profesionáli. Pre tých má Masarykov okruh prichystanú povest' okruhu, kde sa veľmi ťažko hľadá ideálna jazdná stopa.



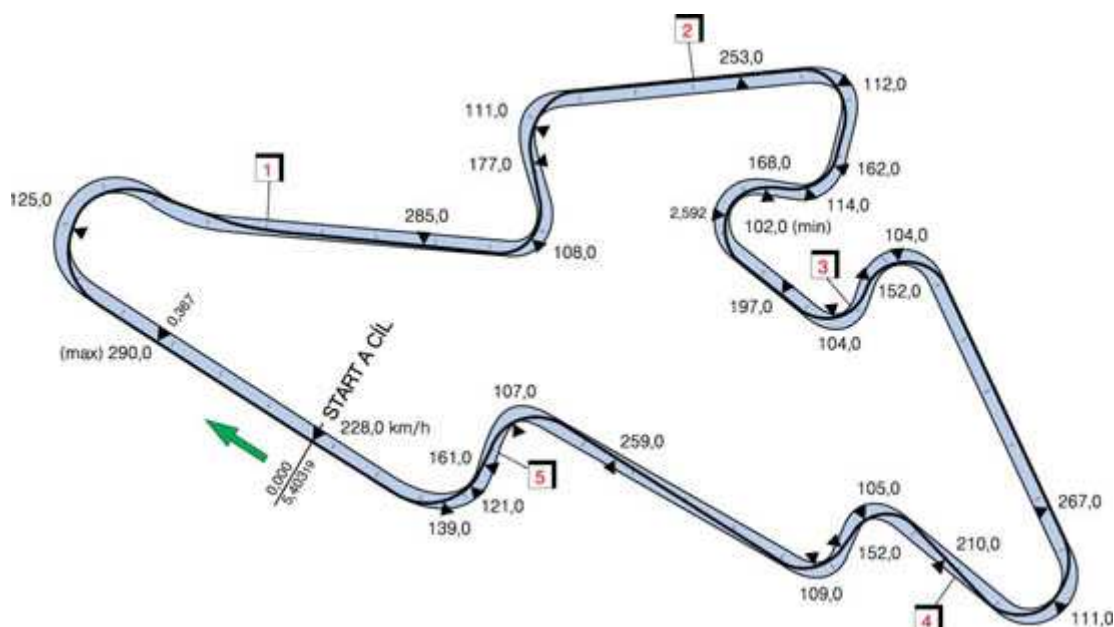
Obr. 4.8 Výškový profil Brnenského závodného okruhu



Oficiálnym traťovým rekordom do doby písania tejto práce je čas 1:47,296, ktorý v roku 2006 stanovil A.Yoong na Formuli A1. Najrýchlejšie kolo pre triedu MotoGP dosiahol v roku 2008 C.Stoner na Ducati a má hodnotu 1:57,199. Z toho istého roku drží M.Fabrizio najlepší čas 1:59,979 taktiež na Ducati, tentoraz ale pre motorky triedy Superbike. V skupine športových vozov triedy GT2, do ktorej by sa mohlo zaradiť aj použité Ferrari F430, drží s vozom Porsche 911 G T3 RS rekordný čas 1:58,193 dvojica Collard/Westbrook. Pre cestné vozidlá triedy S2000 má najlepší čas na BMW 320si F.Porteiro - 2:10,108. V tejto triede by zasa mohol pretekať Renault Mégane. Úplne najrýchlejším a zároveň neoficiálnym traťovým rekordom je čas 1:35,974, ktorý zajazdil v roku 2009 M.Gené na špeciáli Formula 1 2008.

## 4.4.2 Postup tvorby

Pri tvorbe okruhu som postupoval podľa mapky dostupnej na domovskej internetovej stránke okruhu<sup>6</sup> spolu s mapou od spoločnosti Google.



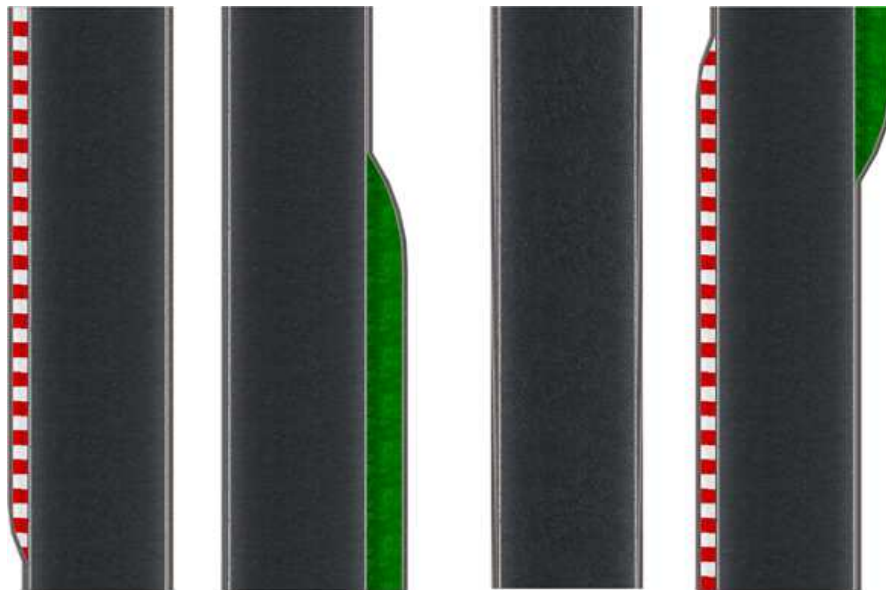
Obr. 4.9 Mapa Masarykovho okruhu v Brne

Celá trať je vytvorená z malých kúskov cesty, ktoré sú pospájané za seba do jedného celku. Stavba okruhu teda pripomína štýl detských autodráh. Toto nám dáva voľnosť pri tvorbe okruhu a v budúcnosti sa môže pomocou týchto stavebných blokov vytvoriť akýkoľvek iný okruh, alebo cesta.

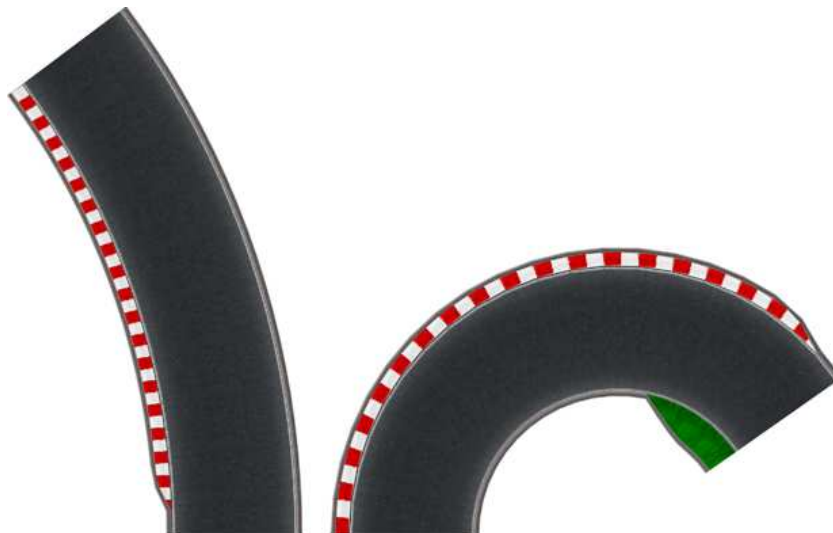
Pred samotnou tvorbou bolo teda treba vytvoriť rôzne úseky, ktoré sa na trati vyskytujú. Rôzne variácie tvoria okrajové časti vozovky ako obrubníky a výjazdové zóny. Celkovo je ich vytvorených trinásť, štyri z nich sú zobrazené na Obr. 4.10. Na všetky bloky bol aplikovaný nástroj Ohyb, ktorý je dostupný v spomínanom modelačnom programe Cinema4D. Pomocou tohto nástroja sa ohýbajú potrebné úseky trate (Obr. 4.11) tak, aby čo najvernejšie kopírovali skutočnú predlohu. Kvôli tomuto ohybu musí mať jeden blok trate asi 40 trojuholníkov. Tento počet je zistený experimentovaním a je vybraný tak, aby bolo dosiahnuté čo najlepšie zaoblenie trate pri čo najmenšom počte polygónov.

<sup>6</sup> <http://www.automotodrombrno.cz/>

Štyridsať trojuholníkov má však iba cesta. Pridaním ďalších okrajových doplnkov narastie tento počet priemerne na 350 trojuholníkov na jeden blok. Na rovné úseky sa samozrejme ohyb neaplikuje a preto sú celé tieto úseky vytvorené iba zo šiestich polygónov.



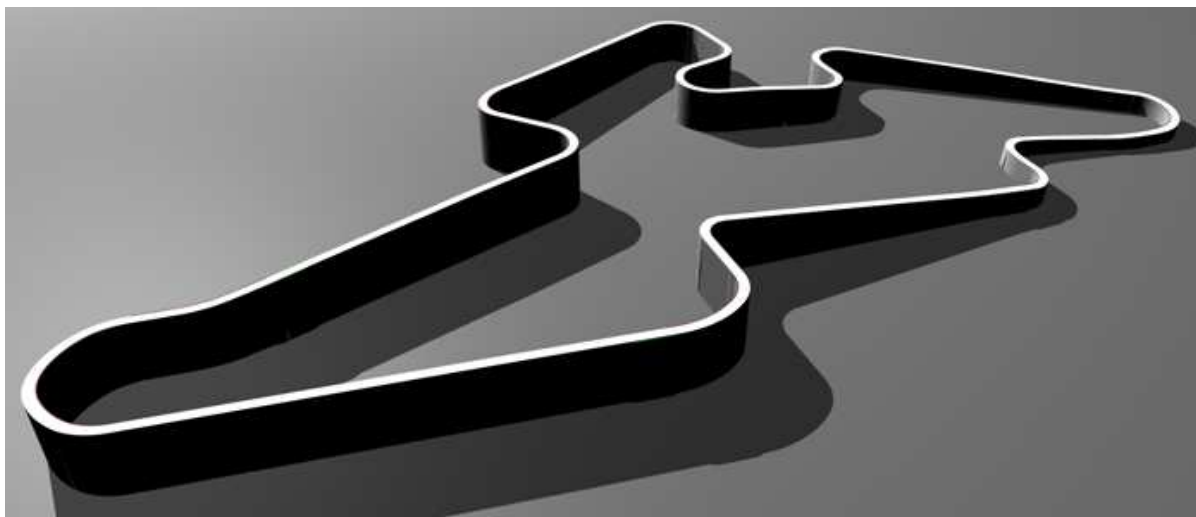
**Obr. 4.10 Pripravené stavebné bloky trate**



**Obr. 4.11 Bloky trate po aplikovaní nástroja Ohyb**

Po celej dĺžke trate je použitých presne 60 stavebných blokov a spolu tvoria objekt s 17900 polygónmi.

Pri vytváraní výškového profilu neboli dostupné také detailné údaje ako pri tvare okruhu, ale podľa Obr. 4.8 som sa snažil profilovú krivku čo najpresnejšie dodržať. Výsledok takto vytvorenej trate vyobrazuje obrázok nižšie.



**Obr. 4.12 Výškový profil vytvorenej trate**

Ďalším krokom bolo doplnenie okolia trate a výbehových zón. Toto okolie som sa, hlavne kvôli kolíznym objektom, rozhodol rozdeliť do dvoch častí. Prvá časť spája trať a ochranné bariéry okruhu. Z tejto sa tak isto vytvorí kolízny objekt aby sa vozidlo po vyjdení z trate neprepadlo. Ochranné bariéry sú taktiež modelované podľa dostupných máp, preto zostali zachované približne reálne veľké výjazdové zóny. Druhá časť okolia už nie je kolízna, lebo hra je robená tak, že za bariéry sa vozidlo nikdy nedostane. Túto časť už nebolo možné vytvoriť podľa originálu, preto je dotvorená podľa osobného estetického cítenia. Aby sa vytvoril dojem, že trať patrí do nejakej krajiny, tak okolo celého okruhu je les, ktorý má za úlohu navodiť pocit ucelenosti krajiny. Na trati sú pre doplnenie detailov umiestnené rôzne objekty ako tribúny, reklamné tabule, rôzne administratívne budovy a parkovisko s autami<sup>7</sup>.



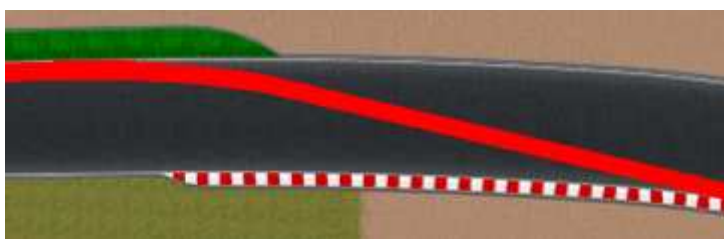
**Obr. 4.13 Finálna podoba okruhu**

<sup>7</sup> Všetky doplnkové objekty sú zdarma a boli stiahnuté z www stránky <http://www.turbosquid.com>

### 4.4.3 Výjazdové zóny a obrubníky

Pri snahe dosiahnuť čo najlepší čas idú pretekári na hranici možností svojich, ale aj vozidla, v ktorom sedia. Takáto jazda vyžaduje pevnú ruku a schopnosť nájsť najideálnejší prejazd trate – teda taký aby bola jazda najkratšia, ale hlavne čo najrýchlejšia. Ideálna stopa mnohokrát vedie na samom okraji trate a preto sa na závodných okruhoch vyskytujú obrubníky a výjazdové zóny.

Výjazdové zóny (na Obr. 4.14 zobrazená zelenou farbou) sa nachádzajú na vonkajšej strane konca zákruty, kde rozširujú trať a tak umožnia rýchlejší a bezpečnejší výjazd zo zákruty. Na Obr. 4.14 je zobrazený koniec zákruty a červenou čiarou je vyznačený ideálny výjazd zo zákruty v smere sprava doľava. Ak by bol pretekár na výjazde o niečo rýchlejší, tak sa ocitne na výjazdovej zóne a stratí prinajhoršom pár desiatín sekundy. Ak by však táto záchranná zóna chýbala, tak by sa dostal mimo trať, kde by mohol stratiť nadvládu nad vozidlom a pravdepodobne by došlo k nehode, čo by znamenalo stratu až niekoľkých sekúnd, ak nie koniec závodu.



Obr. 4.14 Ideálny výjazd zo zákruty

Na vnútornej strane zákrut sa zasa nachádzajú obrubníky. Spravidla sú členité, čo spôsobuje vibrácie a zvukový efekt. Tieto dva signály dávajú pretekárovi najavo, že už sa nachádza na samom okraji zákruty. Obrubníky, ktoré som vytvoril pre túto trať sú zaujímavé tým, že polovica bližšie



Obr. 4.15 Detail obrubníku



k trati je rovná a jej úlohou je rozšíriť trať. Ak sa držia kolesá vozidla na tejto polovici, pretekárovi nič nehrozí. Po druhej polovici obrubníku sa však jazdiť neodporúča. Túto časť tvoria šikmé plôšky, ktoré majú výškový rozdiel až 10cm. Výškový rozdiel nie je okamžitý, ale ako je vidno na Obr. 4.15, postupne narastá od stredu k okraju obrubníka. Koleso, ktoré sa dostane do tejto časti stráca priľnavosť a spôsobuje značné vibrácie. Hlavne pri prejazde rýchlou zákrutou znamená strata priľnavosti veľký problém a vo väčšine prípadov vedie k nekontrolovateľnému šmyku.

## 4.5 Počasie

Najväčšou neznámou pri pripravovaní závodného automobilu na pretek je nadchádzajúce počasie. Hlavne dážď a sneh drasticky ovplyvňujú povrchové vlastnosti vozovky. S pribúdajúcou vlhkosťou klesá priľnavosť cesty a vozidlo sa stáva horšie ovládateľné. Zároveň nie je umožnené kolesám preniesť na cestu toľko výkonu ako na suchú vozovku a teda sú autá aj pomalšie. Nastavenie vozidla na aktuálne počasie spravidla problém nie je. Problém nastáva pri dlhších pretekoch, kde treba zohľadniť aj zmeny v počasí. Samozrejme sú k dispozícii rôzne predpovede počasia, ale tie nikdy nie sú presné a počasie sa môže zmeniť z minúty na minútu. Azda najvýznamnejšie počasím ovplyvňovaný pretek je šampionát Formule 1. Extrémne výkony áut spôsobujú veľmi viditeľné zmeny pri rýchlostiach s pribúdajúcou vlhkosťou trate. Monoposty formule 1 môžu, pre maximalizovanie výkonov a zvýšenie bezpečnosti, obúvať tri hlavné kategórie pneumatík. Na suchej trati musia kolesá preniesť čo najväčšie množstvo výkonu z motora na vozovku. Preto je ich dezén úplne hladký (Obr. 4.16 vľavo). Ak sa tieto pneumatiky, v pretekárskej terminológii nazývané *sliky*<sup>8</sup>, zahrejú na optimálnu teplotu, úplne sa prilepia na trať a vozidlo pevne drží v zákrutách aj pri vysokých rýchlostiach. S pribúdajúcou vlhkosťou trate sa hladké pneumatiky šmýkajú, preto sa na vozidlo nasadia prechodné pneumatiky (Obr. 4.16 v strede). Majú jemný dezén, ktorý odvádza malý objem vody zo stredu pneumatiky a tým sa časť vozovky pod pneumatikou vysuší. Ak je na trati priveľa vody a prechodné pneumatiky nestačia všetku vodu vytlačiť, na rad prichádzajú pneumatiky do extrémneho počasia, ktoré sú zobrazené na Obr. 4.16 vpravo. Ako je vidno majú veľmi členitý dezén, ktorý sa postará o odvádzanie veľkého množstva vody. Dotyčná plocha s vozovkou je však oproti hladkým gumám oveľa menšia a sú schopné preniesť menšie množstvo výkonu.



Obr. 4.16 Pneumatiky pre vozy F1 do rôzneho počasia<sup>9</sup>

<sup>8</sup> z angl. slick - hladký

<sup>9</sup> <http://www.autofieldguide.com/blogs/pirelli-formula-one-lean>,

[http://www.formula1.com/wi/225x/manual/pirelli\\_f1\\_intermediate\\_coppia.jpg](http://www.formula1.com/wi/225x/manual/pirelli_f1_intermediate_coppia.jpg)

Okrem vlhkosti trate pôsobí na výkonnosť vozidla, znova hlavne na schopnosť pneumatík preniesť výkon na vozovku, zmena teploty. Slnečné počasie a vysoká teplota rozpaľujú asfalt na trati, ktorý tak môže dosiahnuť až 50°C. Naopak studený dážď a zatiahnutá obloha môžu spôsobiť pokles teplôt až na hranicu 10-15°C. Z toho jasne vyplýva, že pneumatiky do rôzneho počasia majú rozdielne hodnoty teplôt, kedy dosahujú maximálny výkon. Preto ak sa napríklad zvolí pneumatika do extrémneho počasia príliš skoro a na trati je len menšie množstvo vody, dochádza k prehrievaniu pneumatiky a rýchlemu opotrebeniu a poklesu výkonu.

Aj vďaka veľkému vplyvu počasia na jazdné vlastnosti vozidla som sa rozhodol implementovať jednoduché počasie aj do môjho simulátora. V menu hry je možnosť výberu zo štyroch variant počasia. Primárne je nastavená voľba náhodného počasia.

V tomto móde sa pred štartom závodu náhodne zvolí, či bude zimné obdobie, alebo nie. Podľa obdobia sa vygeneruje teplota vzduchu medzi 0-15°C pre zimné obdobie a pre ostatné 12-32°C. Vygenerovaná teplota sa potom v simulácii používa na výpočet hustoty vzduchu z rovnice (4.6), podľa ktorého sa následne vypočíta odpor vzduchu pôsobiaci na vozidlo zo vzorca (4.7). Rozdiel v tomto režime počasia oproti ostatným je v tom, že sa dynamicky mení počasie počas behu simulácie. Zmena však prebieha v rámci vygenerovaného obdobia a teda sa nemôže stať, že by pršalo a následne snežilo. Zmena počasia je vytvorená tak, že sa vždy, v rozmedzí jednej až dvoch minút, náhodne zvolí intenzita dažďa (snehu). Toto časové rozpätie je také malé kvôli demonštračným účelom. Nie je však problém ho zväčšiť, či posunúť ďalej. Následne sa stanoví čas, kedy dôjde k nasledujúcej zmene a vypočíta sa prírastok intenzity zrážok aby sa v určenom časovom úseku dosiahla plynulá zmena zo súčasného stavu do požadovaného. Ak je prírastok kladný, tak sa intenzita zrážok zvyšuje, ak je záporný, tak sa znižuje. Po ubehnutí stanoveného času sa znova vygeneruje nová hodnota intenzity zrážok a nová doba zmeny počasia. Zároveň so zmenou intenzity zrážok sa plynule mení aj obloha, kde postupne pribúdajú a ubúdajú mračná (Obr. 4.17).



Obr. 4.17 Zmena oblohy so zmenou intenzity zrážok

Ostatné tri módy sú statické. Je to z dôvodu reprezentatívnosti dát, aby sa dali porovnávať zajazdené časy. Ak by bolo počasie dynamické, tak by dochádzalo k veľkým výkyvom pri ovládateľnosti vozidla a porovnanie časov by nebolo presné. Takto sa na základe výdatnosti zrážok určí, či sa jedná o slabé, stredné alebo silné zrážky a porovnávajú sa časy s rovnakou kategóriou sily.

Pri snežení a pršaní je začiatok obdobný s predchádzajúcim prípadom. Sneženie predstavuje stred zimného obdobia a pred štartom sa generuje teplota z intervalu  $-10$  až  $+5^{\circ}\text{C}$ . Pršanie reprezentuje jeseň a tu sa volí teplota vzduchu z rozmedzia  $5-25^{\circ}\text{C}$ . Ďalej sa zvolí intenzita zrážok a na základe toho sa nastaví obloha. Slnečné počasie je zasadené do letného obdobia a preto sa vyberajú teploty od  $20^{\circ}\text{C}$  až po  $40^{\circ}\text{C}$ . U slnečného počasia taktiež dochádza k dynamickým zmenám na oblohy, ale je to iba vizuálny efekt.

Implementovaný model počasia je zjednodušený v tom, že sa nepočíta s prírastkom vlhky na trati, ktorý je reálne spôsobený zrážkami, ale počíta s okamžitou zmenou vlastností trate pri zmene intenzity zrážok. To znamená, že aj pri úbytku zrážok sa ihneď trať „vysušuje“ a teda sa okamžite zlepšujú jazdné podmienky. V závislosti od mohutnosti zrážok sa pri daždi zhorší priľnavosť vozovky maximálne o 22% a pri snežení až o 33%. Tieto hodnoty pravdepodobne nie sú reálne, ale zvolil som ich na základe testovania aby zhruba zodpovedali stavu vozovky.

## 4.6 Vstupné menu

Výsledná aplikácia sa dá spustiť dvomi spôsobmi. Prvým spôsobom je priame spustenie cez súbor *sim.exe*. V tomto prípade sa náhodne vyberie vozidlo a počasie sa nastaví na dynamické. Samozrejme je znemožnené ukladanie časov na internet, lebo sa nevie, ktorý hráč práve hrá.

Druhá možnosť je spustiť priložený spúšťač *Launcher.exe*. Je to samostatná aplikácia vytvorená pomocou knižnice **wxWidgets**, ktorá slúži na nastavenie parametrov a následné spustenie súboru *sim.exe* vrátane zvolených nastavení. Aplikácia pozostáva z troch nastavovacích stránok – úvodná stránka, nastavenie vozidiel a nastavenie trate.

### 4.6.1 Úvodná stránka

Po spustení aplikácie *Launcher.exe* sa zjaví jej úvodná stránka, ktorá primárne slúži na nastavenie hráčovho profilu. Je tu napríklad možnosť zvoliť si meno, ktorým bude užívateľ v hre identifikovaný. Ak má hráč prístup k internetu, môže zadať heslo a kliknutím na tlačidlo *Log in* sa prihlásiť na svoj účet. Ak však ešte nemá vytvorený svoj účet, tak sa automaticky pri prvom prihlásení vykoná registrácia. Každý užívateľ by mal zvážiť prítomné upozornenie o tom, že heslá sa prenášajú nešifrované a preto by nemal v žiadnom prípade používať heslo, ktoré bežne používa. Pri hre, kedy je hráč prihlásený k účtu, sa pri každom meranom úseku porovnávajú časy so vzdialenou databázou a zároveň sa aktualizujú údaje o traťovom rekorde. Veľmi pomalé internetové pripojenie môže spôsobovať nepríjemné zasekávania čakaním na odpoveď z internetu. Preto aplikácia umožňuje spustiť hru s voľbou „pomalé pripojenie“ kedy sa pred začatím hry načíta najlepší čas hráča a celkový najlepší čas. Počas samotného jazdenia aplikácia vôbec s okolitým svetom nekomunikuje, ale pri skončení hry sa znova aktualizujú dáta na externom serveri. Ak sa hráčovi nepodarí zlepšiť svoj najlepší čas, tak sa samozrejme databáza nemení.

Táto stránka ešte umožňuje pridanie druhého hráča do hry, ktorý sa tak isto môže prihlásiť k svojmu účtu. Nie je to však podmienka a je možné aj to, aby bol jeden hráč prihlásený a druhý nie. Okrem toho je tu prítomné zhrnutie ostatných nastavení a tlačidlo pre štart hry.



Obr. 4.18 Úvodná stránka spúšťáča

## 4.6.2 Nastavenie vozidiel

Druhá stránka slúži na výber a nastavenie vozidiel. Obidve dostupné vozidlá sú tu graficky predstavené a sú pri nich vypísané aj základné vlastnosti.

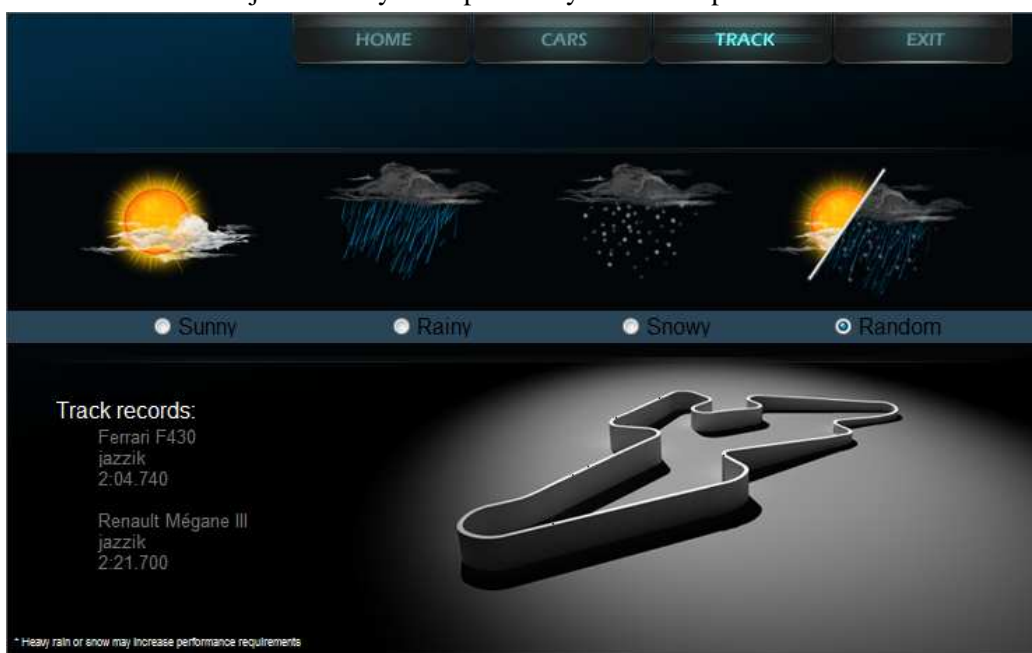
Každý hráč si môže zvoliť auto podľa vlastného uváženia, alebo to môže nechať na náhodu a zvoliť si možnosť *Random*. Ďalej je možné nastaviť prevodovku a to výberom medzi manuálnym radením a automatickým. Nakoniec si môže každý užívateľ zvoliť ľubovoľnú farbu, ktorá bude v hre aplikovaná na jeho vybrané vozidlo.



Obr. 4.19 Stránka s výberom a nastavením vozidiel

### 4.6.3 Nastavenie trate

Na poslednej stránke sú vypísané traťové rekordy pre obidve vozidlá spolu s mapkou okruhu. Tak isto je tu možnosť zvoliť si jednu zo štyroch spomínaných variácií počasia.



Obr. 4.20 Stránka s nastavením počasia na trati

## 4.7 Ovládanie hry

Hra sa v súčasnej podobe dá ovládať iba pomocou klávesnice. Klávesy pre prvého hráča sú:

Akcia:	Hráč č.1	Hráč č.2
Plyn:	šípka hore	e
Brzda:	šípka dole	d
Zabočenie vľavo:	šípka doľava	s
Zabočenie vpravo:	šípka doprava	f
Ručná brzda:	medzerník	ľavý Shift
Reset pozície:	l	r
Zaradenie:	j	q
Podradenie:	m	a
Spojka:	k	w
Ackermanov riadiaci systém:		F1
Reverzný riadiaci systém:		F2
Paralelný riadiaci systém:		F3
Hybridný riadiaci systém:		F4



### 4.7.1 Experimentálne ovládanie hry

Závodné hry – a hlavne závodné simulátory - sa najlepšie ovládajú pomocou volantových a iných analógových ovládačov. Riadené vozidlo presne reaguje na prudkosť a silu otáčania volantom a stláčania plynového, či brzdového pedála. Nie každý má však možnosť zadovážiť si takýto ovládač a preto sa musí uspokojiť s obyčajnou klávesnicou. Pri hraní závodného simulátora na klávesnici treba simulovať analógový vstup opakovaným ťukáním do klávesov, lebo väčšina klávesníc má pre klávesy iba dva stavy – stlačený a nestlačený. Hlavne pri zabáčaní to však spôsobuje neustály pohyb kolies do strán, ktorý vzniká pri stlačení a uvoľnení riadiacich klávesov.

Pri testovaní aplikácie som zistil, že práve toto najviac znehodnocuje dojem z hry. Veľa hráčov sa sťažovalo na prílišnú citlivosť ovládania. Po dlhšom hraní si však veľká väčšina na ovládanie zvykla a aj zajazdené časy sa postupne zrýchľovali. Stále ma však trápil prvotný dojem a preto som sa rozhodol do aplikácie pridať experimentálne riešenie analógového vstupu pomocou ďalších kláves. Úmyslom bolo dosiahnutie toho, aby hráč mohol stále držať šípky a inými tlačidlami si plynule reguloval množstvo aplikovania danej klávesy. Toto ovládanie je prístupné iba v móde jedného hráča a zahŕňa klávesy **1-9** a **q-o**. Pomocou číselných kláves sa dá ovplyvniť uhol natočenia volantu. Ak teda hráč ide do veľmi jemnej pravotočivej zákruty, tak stlačí klávesu **2** a zároveň drží šípku doprava. Tento úkon spôsobí, že sa volant natočí iba do 20% z maximálnej možnej rotácie volantu. Ak by potreboval trochu viac zatočiť, tak môže pustiť **2** a stlačiť klávesu **3**, čo povolí hranicu otočenia volantu na 30%. Obdobne je to pri klávesoch **q-o**, ktoré podobne ovplyvňujú polohu plynového pedála. V tomto prípade stlačené **q** spolu so šípkou hore znamená, že je plynový pedál stlačený z 90%. Prejdením na klávesu **w** sa pedál dvihne do polohy 80%. Smer stúpania percentuálneho podielu pre riadenie a plyn je protichodný z toho dôvodu, že sa počíta s tým, že mierne zákruty sú rýchle a ostré naopak pomalé. Preto pre pohodlnejšie ovládanie umožní kombinácia kláves napríklad **3** a **e** prejazd rýchlou zákrutou s 30percentným vytočením volantu a na 70percent stlačeným plynom.

Táto možnosť ovládania vozidla je stále aktívna, ale hráč môže hrať klasickým spôsobom bez akéhokoľvek obmedzenia. Záleží len na ňom, aký typ ovládania mu vyhovuje.

## 4.8 Použité zdroje

### Zdroje 3D modelov

- 3D model auta Ferrari F430 <sup>10</sup>
- 3D model auta Renault Mégane III <sup>11</sup>
- 3D modely tribún a okolitých budov<sup>12</sup>

### Zdroje použitých tutoriálov a zdrojových kódov

- tutoriál na tvorbu tachometra [13]
- tutoriál na tvorbu pomocnej triedy na odchytyvanie eventov kláves [14]
- rôzne inšpirácie z demo súborov knižníc OpenSceneGraph a Bullet

<sup>10</sup> <http://www.gfx-3d-model.com/2010/05/ferrari-f430-3d-model/>

<sup>11</sup> <http://www.gfx-3d-model.com/2011/01/renault-megane-3d-model>

<sup>12</sup> <http://www.turbosquid.com/>

## 4.9 Zahrievanie pneumatík

Posledná vec, ktorú som mal v pláne zakomponovať do mojej práce bolo zohľadnenie zahrievania pneumatík. Simulácia termodynamických javov na pneumatike by bola veľmi náročná a ani nie je súčasťou zadania práce. Preto som pracoval na veľmi jednoduchom modeli absorbovania a vydávania tepla pneumatikami, ktorý mal zohľadňovať ohrievanie pneumatík pri brzdení. Pneumatiky sa tak isto mali zahrievať pri rýchlych prejazdoch zákrutami a pri šmykoch. Naopak pri rýchlej jazde a pri jazde na vlhkej trati by sa mierne ochladzovali. Od teploty pneumatík mali potom byť závislé aj jazdné vlastnosti vozidla. Studené a prehriate pneumatiky by tak zhoršovali ovládateľnosť auta. Kvôli časovej tiesni som však túto časť nestihol dokončiť a preto vo výslednej aplikácii nie je dostupná. V zdrojovom kóde programu je však rozrobená časť ponechaná, ako komentár, pre prípadný ďalší vývoj aplikácie.

# Záver

V rámci semestrálneho projektu som sa naučil pracovať s knižnicami OpenSceneGraph a Bullet. Hlbšie som prenikol do problematiky dynamiky vozidla [8] a preštudoval niektoré hlavné mechanické procesy vyskytujúce sa u áut. Následne som v letnom semestri vytvoril 3D prostredie pre simuláciu v podobe závodného Masarykovho okruhu v Brne a spracoval aplikáciu do formy závodnej hry. Vytvorený fyzikálny model vozidla je však možné s menšími úpravami použiť v akejkoľvek inej aplikácii založenej na knižniciach OSG a Bullet.

Hra je vytvorená pre jedného, alebo dvoch hráčov cez vertikálne rozdelenú obrazovku. Hráčom je samozrejme umožnené kolidovať medzi sebou, alebo zábranou popri trati. Hra sa môže spustiť priamo cez spustiteľný súbor *sim.exe*, kedy budú aplikované náhodné nastavenia, alebo sa môže spustiť cez samostatnú spúšťačiu aplikáciu *Launcher.exe*, ktorá slúži na presné nastavenie parametrov. Pre väčšiu pútavosť hry som do aplikácie pridal dynamicky meniace sa počasie a možnosť ukladania najlepších zajazdených časov do databázy na vzdialenom serveri. Merané časy sú rozdelené do viacerých kategórií podľa zvoleného auta a počasia. To umožňuje hráčom merať si sily medzi sebou aj s rôznymi nastaveniami. Počasie ovplyvňuje nie len povrchové vlastnosti vozovky a tým aj ovládateľnosť vozidla, ale na základe odpovedajúcej teploty ovzdušia sa vypočíta hustota vzduchu a následne veľkosť odporu vzduchu, ktorý musí vozidlo prekonávať. Pre experimentálne účely som do hry pridal náhradu analógových ovládačov. Hráčovi je umožnené stále držať ovládaciú klávesu a klávesmi 1-9, resp. q-o, plynulo korigovať aplikovanú silu stlačenia. Samozrejme je hru možné ovládať aj bez tohto spôsobu.

Vo výslednej aplikácii som použil zjednodušený model vozidla z Bulletu a rozšíril som ho o vlastný motor a prevodovku, ktorá môže byť manuálna, alebo automatická. K modelu som taktiež vytvoril štyri typy riadiacich mechanizmov, medzi ktorými sa dá počas závodu prepínať. Celý model som obalil funkciami, ktoré umožnia ľahkú integráciu vozidla, resp. vozidiel, s ľubovoľnými parametrami do zvoleného grafu scény OpenSceneGraphu.

Aby sa dala aplikácia nazvať plnohodnotným fyzikálnym simulátorom auta, musí ešte prebehnúť veľa vylepšení, ku ktorým som sa vďaka obmedzenému času nedostal. V práci je rozpracované zohľadňovanie zahrievania pneumatík v závislosti od jazdy, ale táto časť už nebola dokončená, preto je rozpracovaný zdrojový kód v aplikácii uvedený ako komentár. Medzi ďalšie možné vylepšenia fyzikálneho modelu auta a celej aplikácie by sa dali zaradiť:

- Zohľadnenie funkcie diferenciálu, aby prepínal otáčky kolies podľa aktuálneho odporu
- Prepočítanie ťažiska auta, pre zlepšenie ovládateľnosti
- Spresnenie výpočtu brzdného účinku motora
- Interpolovanie výsledkov z ray cast detekcie, alebo
- Zámena Ray cast metódy detekcie kolízií kolies za Convex cast metódu
- Vylepšenie simulácie trenia pneumatík
- Započítanie opotrebovania pneumatík
- Pridanie časticových efektov
- Zanechávanie stôp na ceste pri šmyku
- Deformačný model vozidla
- Sofistikovanejší model počasia
- Zohľadnenie poruchovosti vozidla
- Možnosť sieťového multiplayer módu a mnoho ďalších



# Literatúra

- [1] COUMANS, Erwin. Bullet [online]. 2010-09-20 [cit. 2011-01-10]. Dostupné z WWW: <<http://code.google.com/p/bullet/>>.
- [2] MADDOCK, Kester. Google docs [online]. 2010-08-16 [cit. 2011-01-10]. Vehicle Simulation With Bullet. Dostupné z WWW: <<https://docs.google.com/Doc?docid=0AXVUZ5xw6XpKZGNuZG56a3FfMzU0Z2NyZnF4Zmo&hl=en>>.
- [3] COUMANS, Erwin. Bullet 2.76 Physics SDK Manual [online]. 2010 [cit. 2011-01-10]. Dostupné z WWW: <[http://bullet.googlecode.com/svn/trunk/Bullet\\_User\\_Manual.pdf](http://bullet.googlecode.com/svn/trunk/Bullet_User_Manual.pdf)>.
- [4] OsgBullet [online]. 2010 [cit. 2011-01-10]. Dostupné z WWW: <<http://code.google.com/p/osgbullet/>>.
- [5] OsgBullet [online]. 2010 [cit. 2011-01-10]. Dostupné z WWW: <<http://osgbullet.vesuite.org/>>.
- [6] Game Physics Simulation [online]. 2007-07-02 [cit. 2011-01-10]. Dostupné z WWW: <<http://bulletphysics.org/wordpress/>>.
- [7] Physics Simulation Wiki : Tutorial Articles [online]. 2007 [cit. 2011-01-10]. Dostupné z WWW: <[http://bulletphysics.org/mediawiki-1.5.8/index.php/Tutorial\\_Articles](http://bulletphysics.org/mediawiki-1.5.8/index.php/Tutorial_Articles)>.
- [8] JAZAR, Reza N. Vehicle Dynamics : Theory and Applications. New York (NY) : Springer Science+Business Media, 2208. 1022 s. ISBN 978-0-387-74243-4.
- [9] SETRIGHT, L. J. K.; WARD, Ian. Anatomy of the Motor Car. London : Orbis Books, 1976. Overdrive, s. 93-95.
- [10] REYENGA, Craig. Craig [online]. 2010, 2010-12-26 [cit. 2011-01-10]. Aerodynamic Drag. Dostupné z WWW: <<http://craig.backfire.ca/pages/autos/drag>>.
- [11] OSG [online]. 2007 [cit. 2011-01-10]. Dostupné z WWW: <<http://www.openscenegraph.org/projects/osg>>.
- [12] DITCHBURN, Keith. DirectX Games Programming [online]. 2004 [cit. 2011-01-10]. Direct3D 3D Models. Dostupné z WWW: <[http://www.toymaker.info/Games/html/3d\\_models.html](http://www.toymaker.info/Games/html/3d_models.html)>.
- [13] OVERBEE, John. DevWebPro [online]. 2010-03-12 [cit. 2011-01-10]. Create a Gauge Interface in Photoshop. Dostupné z WWW: <<http://www.devwebpro.com/create-a-speedometer-gauge-interface-in-photoshop/>>.
- [14] MARINO, Peter Wraae; JENSEN, Michael Bach. OsgHelp [online]. 2008 [cit. 2011-01-10]. OsgHelp. Dostupné z WWW: <[http://www.osghelp.com/readarticle.php?article\\_id=9](http://www.osghelp.com/readarticle.php?article_id=9)>.
- [15] Collision detection#A posteriori .28discrete.29 versus a priori .28continuous.29. In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2011-01-10]. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Collision\\_detection#A\\_posteriori\\_.28discrete.29\\_versus\\_a\\_priori\\_.28continuous.29](http://en.wikipedia.org/wiki/Collision_detection#A_posteriori_.28discrete.29_versus_a_priori_.28continuous.29)>.
- [16] SHELQUIST, Richard . Equations - Air Density and Density Altitude [online]. 1998, 2011-03-21 [cit. 2011-05-12]. An Introduction to Air Density and Density Altitude Calculations. Dostupné z WWW: <[http://wahiduddin.net/calc/density\\_altitude.htm](http://wahiduddin.net/calc/density_altitude.htm)>.

# Zoznam príloh

Príloha 1. Vloženie auta do scény

Príloha 2. Inicializácia fyzikálnej scény a tvorba rigid body telesa

Príloha 3. Parametre charakterizujúce vozidlo Aston Martin DBR9

# Príloha 1. Vloženie auta do scény

```
// car object
Vehicle* car;

// player one's car
KeyMap carKeys; // structure for keys
VehicleModel car1Model; // structure for models and collision shapes

// set up keys, models and col.shapes
{
    // set up keys
    carKeys.throttle = osgGA::GUIEventAdapter::KEY_Up;
    carKeys.brake = osgGA::GUIEventAdapter::KEY_Down;
    carKeys.left = osgGA::GUIEventAdapter::KEY_Left;
    carKeys.right = osgGA::GUIEventAdapter::KEY_Right;
    carKeys.handbrake = osgGA::GUIEventAdapter::KEY_Shift_R;
    carKeys.reset = 'p';
    carKeys.clutch = 'o';
    carKeys.gear_up = 'i';
    carKeys.gear_down = 'k';

    // load chassis model
    car1Model.chassisModel = osgDB::readNodeFile("data/aston/aston_dbr9_body.3ds");
    // create chassis collision shape
    car1Model.chassisShape = new btConvexHullShape();
    car1Model.chassisShape = physics-
>createConvexHullShapeFromConvexTriangleMeshShape("data/aston/aston_dbr9_collision.x");

    // load wheel model (one model for all wheels)
    osg::Node* car1Wheel = osgDB::readNodeFile("data/aston/aston_dbr9_wheel.3ds");

    // create PAT for symetry(for wheels on left side)
    osg::PositionAttitudeTransform* rotatePAT = new osg::PositionAttitudeTransform;
    rotatePAT->addChild(car1Wheel);
    // set PAT rotation
    rotatePAT->setAttitude(osg::Quat(3.14,osg::Vec3f(0,1,0)));

    // set up wheels
    car1Model.wheelModel[0] = car1Wheel; // right-front
    car1Model.wheelModel[1] = rotatePAT; // left-front
    car1Model.wheelModel[2] = rotatePAT; // left-rear
    car1Model.wheelModel[3] = car1Wheel; // right-rear

    // create wheel collision shape and assign to wheels
    btCollisionShape* car1WheelShape = new btCylinderShapeX(btVector3(0.169f, 0.322f,
0.322f));
    car1Model.wheelShape[0] = car1WheelShape;
    car1Model.wheelShape[1] = car1WheelShape;
    car1Model.wheelShape[2] = car1WheelShape;
    car1Model.wheelShape[3] = car1WheelShape;
}

// create player one's car object
car = new Vehicle(physicalWorld, osg::Vec3f(2.5,-3,0.1), car1Model, carKeys);

// load car specifications
car->configVehicle("data/aston/aston_dbr9.ini");

// assign chassis and wheel PATs to OSG root
root->addChild(car->getChassisPAT().get());
root->addChild(car->getWheelPAT(0).get());
root->addChild(car->getWheelPAT(1).get());
root->addChild(car->getWheelPAT(2).get());
root->addChild(car->getWheelPAT(3).get());

// create matrix transform for camera manipulation
osg::MatrixTransform* car1MT = new osg::MatrixTransform();
car->getChassisPAT()->addChild(car1MT);

// set player's name
car->name = "Iceman";
```

## Príloha 2. Inicializácia fyzikálnej scény a tvorba rigid body telesa

```
// init physics
btDynamicsWorld* Physics::initPhysics(){

    // default configuration
    btDefaultCollisionConfiguration* collisionConfig = new
btDefaultCollisionConfiguration();

    // collision dispatcher
    btCollisionDispatcher* dispatcher = new btCollisionDispatcher( collisionConfig );

    // solver
    btConstraintSolver* solver = new btSequentialImpulseConstraintSolver;

    // set-up world min and max
    btVector3 worldAabbMin( -10000, -10000, -10000 );
    btVector3 worldAabbMax( 10000, 10000, 10000 );

    // set-up interface
    unsigned short maxHandles = 1000;
    btBroadphaseInterface* interfce = new btAxisSweep3( worldAabbMin, worldAabbMax,
maxHandles);

    // create dynamic world
    btDynamicsWorld* dynamicWorld = new btDiscreteDynamicsWorld( dispatcher, interfce,
solver, collisionConfig );

    // set gravity
    dynamicWorld->setGravity( gravity );

    return dynamicWorld;
};

/*-----*/

// create rigid body
btRigidBody* Physics::createRigidBody(btDynamicsWorld* world, float mass, const btTransform&
startTransform, btCollisionShape* shape){

    bool isDynamic = true;
    // if mass is 0, rigid body is static, otherwise dynamic
    if(mass == 0.0f)
        isDynamic = false;

    // set local inertia
    btVector3 localInertia(0, 0, 0);

    // calculate inertia if rigid body is dynamic
    if(isDynamic)
        shape->calculateLocalInertia(mass, localInertia);

    // using motion state
    btDefaultMotionState* motionState = new btDefaultMotionState(startTransform);

    // create body
    btRigidBody* body = new btRigidBody(mass, motionState, shape, localInertia);

    // add body to world
    world->addRigidBody(body);

    return body;
};
```

# Príloha 3. Parametre charakterizujúce vozidlo Aston Martin DBR9

Súbor `data/aston/aston_dbr9.ini`:

```
#center of right front wheel
frontWheelChassisConnectionPoint 0.818 1.384 0.327 #x z y
#center of right rear wheel
rearWheelChassisConnectionPoint 0.835 -1.351 0.327 #x z y

#maximum length of the front suspension
frontSuspensionRestLength 0.18 #m
#maximum length of the rear suspension
rearSuspensionRestLength 0.18 #m
#maximum front suspension compression
maxFrontSuspensionTravel 7.0 #cm
#maximum rear suspension compression
maxRearSuspensionTravel 7.0 #cm

#front wheel radius
frontWheelRadius 0.338 #m
#rear wheel radius
rearWheelRadius 0.338 #m
#front wheel width
frontWheelWidth 0.268 #m
#rear wheel width
rearWheelWidth 0.268 #m

#stiffness constant for the front suspension (10.0 - offroad, 50.0 - sports car, 200.0 - f1)
frontSuspensionStiffness 45.0
#stiffness constant for the rear suspension (10.0 - offroad, 50.0 - sports car, 200.0 - f1)
rearSuspensionStiffness 45.0
#damping compression constant for the front suspension (0.0 - undamped, 1.0 - critical damping)
frontDampingCompressionConstant 0.4 # <0,1>
#damping compression constant for the rear suspension (0.0 - undamped, 1.0 - critical damping)
rearDampingCompressionConstant 0.4 # <0,1>
#damping relaxation constant for the front suspension (0.0 - undamped, 1.0 - critical damping), should be slightly larger than frontDampingCompressionConstant
frontDampingRelaxationConstant 0.5 # <0,1>
#damping relaxation constant for the rear suspension (0.0 - undamped, 1.0 - critical damping), should be slightly larger than rearDampingCompressionConstant
rearDampingRelaxationConstant 0.5 # <0,1>

#mass of the vehicle
mass 1100.0 #kg
#chassis length
length 4.697 #m
```

```

#chassis width
width 2.024 #m
#chassis height
height 1.446 #m

#vehicle drive
drive 1 # 0-FWD, 1-RWD, 2-4WD

#gear ratios (1-x)
gearRatios 3.723 2.669 1.680 1.151 0.744
#reverse ratio
reverseRatio 3.333
#differential ratio
differentialRatio 3.1

#coefficient of drag
dragCoefficient 0.34

#max braking force
maxBrakingForce 150 #N
#max handbrake force
maxHandbrakeForce 1370 #N

#engine performance
dynoGraph{
#rpm power torque
3100 90 150
3300 116 185
4000 183 240
4500 266 310
4900 359 385
5000 378 397
5200 398 403
5300 403 400
5500 408 390
6000 417 365
6500 423 342
6700 425 333
7000 403 302
}

```